

# Evolving Fuzzy Systems from Data Streams in Real-Time

Plamen Angelov, *Senior Member IEEE*, and Xiaowei Zhou, *Student Member IEEE*

**Abstract**— An approach to real-time generation of fuzzy rule-based systems of eXtended Takagi-Sugeno (xTS) type from data streams is proposed in the paper. The xTS fuzzy system combines both zero and first order Takagi-Sugeno (TS) type systems. The fuzzy rule-base (system structure) evolves starting ‘from scratch’ based on the data distribution in the joint input/output data space. An incremental clustering procedure that takes into account the non-stationary nature of the data pattern and generates clusters that are used to form fuzzy rule based systems antecedent part in on-line mode is used as a first stage of the non-iterative learning process. This structure proved to be computationally efficient and powerful to represent in a transparent way complex non-linear relationships. The decoupling of the learning task into a non-iterative, recursive (thus computationally very efficient and applicable in real-time) clustering with a modified version of the well known recursive parameter estimation technique leads to a very powerful construct – evolving xTS (exTS). It is transparent and linguistically interpretable. The contributions of this paper are: i) introduction of an adaptive recursively updated radius of the clusters (zone of influence of the fuzzy rules) that learns the data distribution/variance/scatter in each cluster; ii) a new condition to replace clusters that excludes contradictory rules; iii) an extended formulation that includes both zero order TS and simplified Mamdani multi-input-multi-output (MIMO) systems; iv) new improved formulation of the membership functions, which closer resembles the normal Gaussian distribution; v) introduction of measures of clusters quality that are used to form the antecedent parts of respective fuzzy rules, namely their age and support; vi) experimental results with a well known benchmark problem as well as with real experimental data of concentration of exhaust gases (NOx) in on-line modeling of car engine test rigs.

## I. INTRODUCTION

OVER the last two decades, the fast development of information technology has not only led to an enormous increase of the speed of computers, but also to enormous amount of data produced and being produced with a very fast rate. Examples of data overload are the Internet, genome information, complex industrial processes etc. [1]. Such high-volume, non-stationary *data streams* bring new challenges to the well established statistical learning methods [2]. In particular, data streams cannot be analyzed in a batch mode, since storing the complete data is often practically impossible. Instead, systems have to be developed that extract tractable *knowledge* from the data in *real-time*.

According to a widely accepted definition, *knowledge*

*generation* or *discovery* refers to the non-trivial process of identifying valid and understandable/interpretable *structure* in the data [1]. From control point of view, this equates to the system structure identification, a process that is usually ignored, while the attention is paid to the system parameter identification (tuning, adjustment, adaptation). Since the data are non-stationary it is logical to assume the *structure* of the data to be also dynamic, that is, to *evolve*.

Very often tuning and adjustment are called ‘adaptation’ [3], ‘self-organisation’ [4], [5] or even ‘evolution’ [6]. It should be noted that all of the above mentioned approaches assume the fuzzy rule-based system structure to be fixed. The Oxford Dictionary gives the following definition for ‘*evolve*’: ‘*unfold; develop; be developed, naturally and gradually*’ [7, p.294]. Contrast this to the more general ‘*evolutionary*’ [same page and source] ‘development of more complicated forms of life (plants, animals) from earlier and simpler forms’, which is naturally related to the ‘*genetic*’ [7, p.358] ‘branch of biology dealing with the heredity, the ways in which characteristics are passed on from parents to offspring.’

We use further the term ‘*evolving*’ fuzzy systems in the sense of ‘gradual development’ of the fuzzy systems structure (rule-base) and their parameters. This new paradigm introduced for neural networks in 1990s [8] and for fuzzy rule-based systems in 2001 [9]-[11] can be regarded as a *higher level adaptation*. Indeed, conventional adaptive systems known from control and system theory [12] deal predominantly with parameter adaptation of linear systems. By comparison, so called evolutionary algorithms (genetic algorithms [13], genetic programming [14] etc.) mimic the evolutionary processes that take place in **populations** of individuals and use operators based on paradigms such as ‘crossover’, ‘mutation’, ‘selection’, ‘recombination’ of chromosomes as mechanisms of adaptation. The emerging *evolving* fuzzy and neuro-fuzzy systems paradigm [9]-[11],[15]-[25] mimics the evolution of **individuals** in nature during their life-cycle, specifically the autonomous mental development typical of humans: learning from experience, inheritance, *gradual* change, knowledge generation from routine operations, rules extraction from the data. A trivial analogy is the way people learn during their life – starting with an empty rule-base they learn new rules during their life from experience and based on the data streams that their preceptors generate to the brain. The development of the rule-base is gradual, but the rules are not fixed or pre-defined. We generate new rules when new facts that can not be described by the existing rules and when they are descriptive enough, not to be ‘one-off’ outliers.

Both authors are with the Intelligent Systems Research Laboratory, Department of Communication Systems, InfoLab21, South Drive, Lancaster University, Lancaster, LA1 4WA, UK; phone +44 (1524) 510391; fax: +44 (1524) 510489, e-mail: [p.angelov@lancaster.ac.uk](mailto:p.angelov@lancaster.ac.uk)

It is well known that fuzzy rule-based (FRB) systems are universal function approximators [27]; they are suitable for extracting interpretable knowledge therefore, they are a promising framework for designing effective and powerful prognostic, classification, and control systems [9]-[11],[15]-[24].

Traditionally, FRB systems' learning has been addressed for the batch (off-line) case and it was assumed to be fixed. The attention has been shifted towards the evolving design of FRB systems since 2001 [9]-[11], [26]. Several papers has been published that interpret and improve the evolving fuzzy rule-based (eR) systems approach [9]-[10] and the fully recursive version specific to Takagi-Sugeno models, called eTS (from evolving Takagi-Sugeno models) [11]. The present paper can be considered as another contribution to the development of this concept that has its roots in the papers and the monograph published in the beginning of the current century [9]-[11]. Note, that [11] was submitted in 2002 and [10] was submitted in 2001.

One can group the FRB learning methods into two broad categories [1]:

- 1) direct (single phase) learning;  
Supervised learning addresses the identification task as a non-linear optimization problem that is solved numerically [3],[5],[6];
- 2) indirect (two phases) learning;  
this approach presumes initial data partitioning (phase A) using unsupervised data clustering methods [9]-[11],[15]-[20] and system parameter identification (phase B) using supervised learning method such as recursive least squares, RLS.

According to the *evolving systems* paradigm [9]-[10], the structure of the FRB is not fixed, it gradually *evolves* (can expand or shrink). In this paper, the spreads of the membership functions of the fuzzy sets that also represent the projections of the radius of the zone of influence of the clusters onto the inputs axes are considered to be adaptive, recursively updated, too. The quality of the clusters and fuzzy rules respectively in terms of their *age* and *support* can be recursively calculated on-line and used to automatically revise the relevance of the rules. Quality of the clusters/rules parameters *age* and *support* are an additional leverage to update the rule base.

The proposed eXTS FRB system has been tested on a typical benchmark problem of Mackey-Glass chaotic time-series prediction as well as on real data for designing an intelligent sensor for exhaust gases (NOx) concentration real-time estimation in car engine tests.

## II. THE STRUCTURE OF THE EXTENDED TAKAGI-SUGENO MODEL

We consider an extended case of Takagi-Sugeno models (xTS) that are multi-input-multi-output (MIMO) [16] and can be described as a set of fuzzy rules of the following form:

$$R^i : IF(x_1 \text{ is close to } x_1^{i*}) AND \dots AND(x_n \text{ is close to } x_n^{i*})$$

$$THEN(y^i = f^i) \quad (1)$$

where  $R^i$  denotes the  $i^{th}$  fuzzy rule;  $i=[1,N]$ ;  $N$  is the number of fuzzy rules;  $x = [x_1, x_2, \dots, x_n]^T$  is the input vector;  $(x_j \text{ is close to } x_j^{i*})$  denotes the  $i^{th}$  fuzzy sets of the  $j^{th}$  fuzzy rule;  $j=[1,n]$ ;  $x^{i*}$  is the focal point of the  $i^{th}$  rule antecedent;  $y^i = [y_1^i, y_2^i, \dots, y_m^i]^T$  is the  $m$ -dimensional output of the  $i^{th}$  linear sub-system.

Note that the type of the fuzzy rule depends on the type of the consequent:

- a) It is of first order TS type [28] when the consequents are linear:

$$f^i = x_e^T \pi^i, \quad x_e^T = [1, x^T] \quad (2a)$$

$$\pi^i = \begin{bmatrix} \alpha_{01}^i & \alpha_{02}^i & \dots & \alpha_{0m}^i \\ \alpha_{11}^i & \alpha_{12}^i & \dots & \alpha_{1m}^i \\ \dots & \dots & \dots & \dots \\ \alpha_{n1}^i & \alpha_{n2}^i & \dots & \alpha_{nm}^i \end{bmatrix}$$

where  $\alpha_{nm}^i$  are the parameters of the  $m$  local linear sub-systems

- b) It is of zero order TS type when the consequents are singletons (crisp scalar values):

$$f^i = a^i \quad (2b)$$

where  $a^i = [\alpha_{01}^i \quad \alpha_{02}^i \quad \dots \quad \alpha_{0m}^i]^T$  are the parameters of the  $m$  local linear sub-models.

Note that equations (1) and (2b) describe sM model while the conventional Mamdani type fuzzy model assumes fuzzy consequents [29] apart from the trivial case when several consequents are identical (then these consequents count once in the sM and several times in the zero-order TS model).

The overall output of the eXTS,  $y$  is formed as a collection of loosely/fuzzily combined multiple simpler sub-systems,  $y^i$ . The degree of activation of each rule is proportional to the level of contribution of the corresponding sub-system to the overall output of the eXTS.

$$y = \sum_{i=1}^N \lambda^i y^i \quad (3)$$

where  $y^i$  represents the output of the  $i^{th}$  sub-model;  $\lambda^i$  is the normalized activation level of the  $i^{th}$  rule;  $t^i$  is the firing level of the  $i^{th}$  rule:

$$\lambda^i = \frac{\prod_{j=1}^n \mu_j^i(x_j)}{\sum_{j=1}^n \prod_{j=1}^n \mu_j^i(x_j)} \quad (4)$$

where  $\mu_j^i$  is the membership value of the  $j^{th}$  input  $x_j$ ,  $j=[1,n]$ , to the  $i^{th}$  fuzzy set,  $i=[1,N]$ ;

The membership function can be of any form. The Gaussian bell function is widely accepted due to its generalization capabilities (it resembles normal distribution and covers the

whole domain of the variables, thus avoids potential computational problems):

$$\mu_j^i = e^{-\frac{\|x - x^{i*}\|_j^2}{2(r_j^i)^2}} \quad (5)$$

where  $(r_j^i)^2$ ,  $i=[1,N]$   $j=[1,n]$  is the spread of the membership function, which also represents the radius of the zone of influence of the cluster/rule.

Note that this formulation of the membership function differs from the one used in the off-line subtractive clustering [30] and its on-line version [11] and is closer to the normal Gaussian distribution (to the extent of a constant). In section III a new formula for recursive learning of the spread/radius is introduced as well as a new condition for replacement of the clusters that is formulated over the input data space only and thus avoids contradictory rules contributing to the improvement of the interpretability of the model.

### III. EVOLVING XTS FROM DATA IN REAL-TIME

#### A. Phase A: partitioning input/output data space

Each one of the sub-systems of the exTS operate in a certain sub-area of the input/output data space,  $z = [x^T; y^T]^T$ ;  $z \in \mathbb{R}^{n+m}$ . To identify these regions one can employ real-time clustering thus effectively learning the antecedent part of the fuzzy rules (1)-(2). Some of the published approaches cluster the input data only [20],[22], but in order to find regions of similar functional input-output dependence,  $y=f(x)$  the joint input/output data space needs to be considered.

Two parameters are needed to define a membership function of the type (5), namely the focal point,  $x^{i*}$  and the spread,  $r_j^i$ . If locate the focal points of the rules,  $x^{i*}$  at the cluster centre (note, only coordinates for the inputs are used to define the focal point although coordinates of the outputs are also used in the clustering) and if determine the spread,  $r_j^i$  based on the data the antecedent part of the fuzzy rules are defined.

The real-time evolving clustering approach (eClustering) used in this paper stems from the well known subtractive clustering [30] and Mountain clustering approaches [31]. This method has the following specific features that separate it from the other clustering approaches:

- ✓ It is non-iterative (no search is involved);
- ✓ It has very low memory requirements, because recursive calculations are used;
- ✓ It is fully unsupervised in the sense that number of clusters are not pre-defined (they are determined based on the data density alone);
- ✓ it can start 'from scratch' from the very first data sample assumed to be the first cluster centre;
- ✓ changes of the cluster number and parameters are gradual, incremental, not abrupt

- ✓ it is problem-dependant-parameters free and the radius of influence of the clusters is learning adaptively from the data distribution/ variance per cluster.

This clustering method is based on the recursive calculation of the value called *potential* [11]. Potential calculated at a data point is a function of accumulated proximity which represents the *density* of the data surrounding this data point:

$$P_k(z_k) = \frac{1}{1 + \left( \sum_{j=1}^{n+m} \sum_{i=1}^{k-1} \|z_i - z_k\|_j^2 \right) / (k-1)} \quad (6)$$

where  $P_k(z_k)$  denotes the potential of the data point ( $z_k$ ); calculated at time  $k$  starting from  $k=2$ ;  $n+m$  defines the dimensionality of the input/output data space ( $z \in \mathbb{R}^{n+m}$ )

This approach is prototype-based (some of the data points are used as prototypes of cluster centers). A number of algorithms for neural networks and neuro-fuzzy systems learning (such as GWRNN [8], GAP-RBF [25], SONFIN [24], EFuNN [21], SOFNN [22]) published recently use *mean-based* clustering. The centers are located at the *mean*, which, in general, does not coincide with any data point. Therefore, these approaches usually form large number of clusters that one needs later to 'prune' [25]. These clustering approaches are *thresholds*-based and the result highly depends on the selection of the appropriate threshold(s).

The clustering procedure starts 'from scratch' assuming that the first data point available is a centre of a cluster. This assumption is temporary and if *a priori* knowledge exists the procedure can start with an initial set of cluster centers that will be further refined. The coordinates of the first cluster centre are formed from the coordinates of the first data point ( $x^{1*} \leftarrow x_1$ ). Its potential is set to the ideal value,  $P_1(z^{*1}) \rightarrow 1$ .

Starting from the **next** data point which is read in real-time, the following steps are performed:

- 1) calculate its potential,  $P_k(z_k)$ ;
- 2) update the potential of the existing cluster centers (by definition their potential will be affected by adding a new data point);
- 3) calculate the increment of the potential the new data point brings in respect to all previously existing centers. Based on this comparison and the membership to the existing clusters take one of the following actions:
  - a) **add** a new cluster center based on the new data point;
  - b) **remove** a cluster that describes well the new point that brings an increment in the potential; **replace** it with a cluster formed around the new point;
  - c) **ignore** (do not change the cluster structure).

These basic steps are repeated for the next data sample

( $k \leftarrow k+1$ ) until there is no more available data or until a requirement to stop the process is received.

**Step 1)** includes the **recursive** calculation of the potential of the new data point as described in [11] by:

$$P_k(z_k) = \frac{1}{1 + \alpha_k + (\beta_k - 2\gamma_k)/(k-1)} \quad (7)$$

$$\alpha_k = \sum_{j=1}^{n+m} (z_k^j)^2; \quad \beta_k = \sum_{i=1}^{k-1} \sum_{j=1}^{n+m} (z_i^j)^2; \quad \gamma_k = \sum_{j=1}^{n+m} z_k^j \Gamma_k^j; \quad \Gamma_k^j = \sum_{i=1}^{k-1} z_i^j$$

Where

Values  $\alpha_k$  and  $\gamma_k$  represent accumulated projections in each dimension of the data space and can easily be calculated based on the availability of the current data point,  $z_k = (z_k^1, z_k^2, \dots, z_k^{n+m})^T$  only. The values  $\beta_k$  and  $\Gamma_k^j$  require accumulation of past information. This can, however, be stored in two variables only (the scalar,  $\beta_{k-1}$  and the

$(n+m)$ -dimensional vector-column  $\Gamma_k = (\Gamma_k^1, \Gamma_k^2, \dots, \Gamma_k^{n+m})^T$ ).

Using these  $(n+m+1)$  values stored from the previous calculations one can recursively calculate the accumulated quantities  $\beta_k$  and  $\Gamma_k^j$ ; thus the potential of the current point from (7) without memorizing all of the previous points:

$$\beta_k = \beta_{k-1} + \alpha_{k-1}; \beta_1 = 0; \quad (8)$$

$$\Gamma_k^j = \Gamma_{k-1}^j + z_{k-1}^j; \Gamma_1^j = 0 \quad (9)$$

**Step 2)** The potential of a centre ( $z^{i*}$ ) can be calculated recursively as described in [11]:

$$P_k(z^*) = \frac{(k-1)P_{k-1}(z^*)}{k-2 + P_{k-1}(z^*) + P_{k-1}(z^*) \sum_{j=1}^{n+m} \|z^* - z_{k-1}^j\|^2} \quad (10)$$

**Steps 3)** It should be noted that originally this condition was borrowed and modified from the off-line subtractive clustering approach [30] and was formulated in [9],[11] as:

$$\frac{P_k(z_k)}{\max_{l=1}^N P_k(z^{l*})} + \frac{\delta_{\min}}{r} \geq 1 \quad (11)$$

where  $\delta_{\min}$  represents the distance to the nearest cluster centre.

Different extensions of this condition were proposed later by Victor and Dourado and published in [18].

In the present paper we use the following formulation of this step of the algorithm:

**IF** (the new data point brings an increment in the potential in respect to already existing centers:

$$\Delta P_i \begin{cases} > 0; \forall i \\ < 0; \forall i \end{cases}; i = [1, N] \quad (12)$$

**THEN** (a new rule is created around this point):

$$x^{*(N+1)} \leftarrow x_k \quad (13)$$

where  $\Delta P_i = P_k(z_k) - P_k(z^{i*})$

If condition (12) does not hold we do not change cluster structure. If condition (12) holds and this point is described well (membership of  $1/3^{\text{rd}}$  which resembles one sigma region in Gaussian distribution) by already existing centers:

$$\exists i, i = [1, N]; \quad \mu_i^j > 1/3 \quad \forall j, j = [1, n] \quad (14)$$

then we **replace** this cluster by this new point (refresh the centre),  $x^{i*} \leftarrow x_k$

Note that the original approach [9]-[11] and its modifications [15]-[18] consider a constant radius/spread,  $r$ . In [16] the spread is assumed to be different per input variable  $r^j$  but were fixed, not adapting with the data. Additionally, note that  $r$  is used in the formulation of the membership functions (equation (5)) which concern inputs only while the distance;  $\delta_{\min}$  is over the joint input-output data space. We argue that if consider input data space only for the condition (14) contradictory rules will be avoided. If consider the distance in the joint input-output data space as a criteria for replacing the clusters rules that have similar antecedent parts but very different consequents parts may become cluster centers, which will lead to a rule-base with contradictions. Condition (14) replaces such rules. The presumption is that it is better not to allow contradictory rules to be created at the point of entry to the rule-base instead of allowing them at first instance and then simplifying the rule-base.

## B. Quality measures of the generated clusters

In a real-time application only cluster centers are kept in the memory, while all the other data points are discarded. The question arises ‘how well these centers represent the data that were discarded from the memory?’. One way to address this issue is by monitoring properties of the clusters that are formed. These include their *radius* (zone of influence), *support* and *age*.

Support of the cluster (rule) is simply the number of data points that are in the zone of influence of that cluster/rule. It can be determined by assigning each data point at the moment it is first read to the nearest cluster:

$$S^l \leftarrow S^l + 1 \quad \text{for } l = \underset{i=1}{\operatorname{argmin}}^N \|z_k - z^{i*}\| \quad (15)$$

where  $S^l$  is the *support* of the  $l^{\text{th}}$  cluster;  $l = [1, N]$

Due to the incremental nature of the approach the relevance of a cluster (respectively fuzzy rule) may change. One of the measures of the relevance of the rule is its *support* and especially the ratio of the *support* in terms of the overall number of data points available at a moment of time,  $k$ . One can introduce a simple rule for ignoring (practically, removing) fuzzy rules that have very low support:

$$\text{IF}(S_k^i / k < \varepsilon) \text{ THEN } (\lambda^i \leftarrow 0) \quad (16)$$

This rule is *optional*. The value of the threshold  $\varepsilon$  represents the proportion of the data assigned to the  $i^{\text{th}}$  cluster/rule.

In this paper a more efficient and more flexible recursive formulation of the radius is introduced that is more realistic. Indeed, in real cases, the spatial distribution of the samples is difficult to be pre-estimated and it is time varying.

A recursive formula for adaptive calculation of the radius based on the local spatial density is introduced as follows:

$$r_{jk}^l = \rho r_{j(k-1)}^l + (1-\rho) \sigma_{jk}^l; r_{j1}^l = 1 \quad l = \underset{i=1}{\operatorname{argmin}} \|z_k - z^{i*}\| \quad (17)$$

where  $\rho$  is a constant that regulates the compatibility of the new information with the old one. Value of  $\rho=1/2$  means that the new information is as valuable as the existing one. This value gives very good performance results and can be adopted as a non problem-dependent, generic constant;  $\sigma_{jk}^l$  is the local scatter over the input data space [15] that resembles the variance:

$$\sigma_{jk}^i = \sqrt{\frac{1}{S_k^i} \sum_{l=1}^{S_k^i} \|x^{i*} - x_l\|_j^2}; \sigma_{j0}^i = 1 \quad (18)$$

When a new cluster/rule is added,  $N \leftarrow N+1$ , its local scatter [15] is initialized based on the average of the local scatters for the existing clusters (rules):

$$\sigma_{jk}^{N+1} = \frac{1}{N} \sum_{i=1}^N \sigma_{jk}^i, j=[1,n] \quad (19)$$

These measures of the cluster quality can be used for real-time management of the rule-base and as an additional decision when generating new clusters/fuzzy rules.

Note that the spreads of the fuzzy membership functions,  $r_j^i$  (respectively the radius of the clusters) are different for different inputs,  $j=[1,n]$  as well as for different fuzzy rules,  $i=[1,N]$ . This makes possible to define not just hyper-spherical, but also hyper-ellipsoidal clusters that recursively adapt their shape to the spatial information brought in by new data samples. This is illustrated in Fig. 2 as opposed to Fig. 1.

Another parameter that describes the properties of the clusters is the so called *age* [15]. It can be defined as the

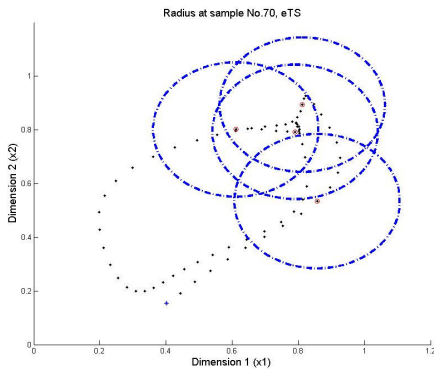


Fig. 1. Cluster radius (membership function spread) for the Mackey-Glass benchmark problem using eTS [11]

number of data samples minus the average sum of the time indices of the data samples:

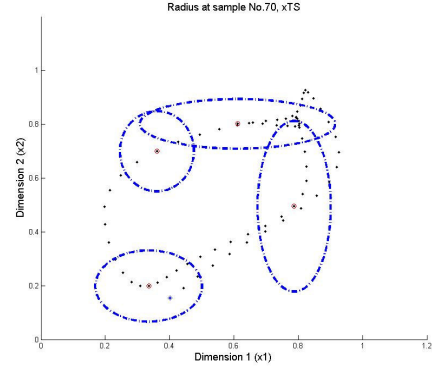


Fig. 2. Cluster radius (membership function spread) for the Mackey-Glass problem;  $r$  are different for different  $i=[1,N]$  and  $j=[1,n]$

$$\operatorname{age}_k^l = k - \frac{2A_k^l}{(k+1)}; l=[1,N] \quad (20)$$

where  $A_k^l = \sum_{i=1}^{N^l} I_i$  denotes the accumulated time of

arrival;  $I_i$  denotes the index of the  $i^{\text{th}}$  data sample (the time instant when this data point was read).

The accumulated time of arrival is calculated in a similar way to the *support* (16), but in respect to the indices:

$$I_k^l \leftarrow I_k^l + k \quad \text{for } l = \underset{i=1}{\operatorname{argmin}} \|z_k - z^{i*}\| \quad (21)$$

In the extreme case, when all of the data samples go into the same cluster/rule, the  $A$  is determined as a sum of the arithmetic progression:

$$A_k = 1 + 2 + \dots + k = \frac{(k+1)k}{2} \quad (22)$$

The *age* of the cluster has values in the range  $(0;k]$  with values close to 0 meaning that recent data is included in this (*young*) cluster and values close to  $k$  meaning that no recent data is included in this (*old*) cluster. *Old* clusters can be replaced by data that have high increment of the potential.

### C. Phase B: Learning consequent part's parameters

Once the antecedent part of the fuzzy model is determined and fixed the identification of parameters of the consequent part,  $\pi^j$  can be solved as a RLS estimation problem as detailed in [11]. It should be noted that the real-time algorithm must perform both tasks (data partitioning and parameter estimation) at the same time instant (per data point) for a time significantly shorter than the sampling period.

In this way, the antecedent part of the rules can be determined in a fully unsupervised way, while the consequent part requires a supervised feedback. The supervision is in the form of error feedback which guarantees optimality (subject to fixed rule base structure) of the parameters of the consequent part.

The overall output can be given in a vector form [11]:

$$y = \psi^T \theta \quad (23)$$

where  $\theta = [(\pi^1)^T, (\pi^2)^T, \dots, (\pi^N)^T]^T$  is a vector formed by the sub-system parameters;  $\psi = [\lambda^1 x_e^T, \lambda^2 x_e^T, \dots, \lambda^N x_e^T]^T$  is a vector of the inputs that are weighted by the normalized activation levels of the rules,  $\lambda^i$ ,  $i=[1, N]$  for the first order TS system (2a) and  $\psi = [\lambda^1, \lambda^2, \dots, \lambda^N]^T$  for the sM system, (2b).

The optimal in least squares sense solution can then be found applying weighted RLS, wRLS as detailed in [11]:

$$\hat{\theta}_k = \hat{\theta}_{k-1} + C_k \psi_k (y_k - \psi_k^T \hat{\theta}_{k-1}) \quad (24)$$

$$C_k = C_{k-1} - \frac{C_{k-1} \psi_k \psi_k^T C_{k-1}}{1 + \psi_k^T C_{k-1} \psi_k}; k=2, 3, \dots \quad (25)$$

where  $\hat{\theta}_1 = 0$ ;  $C$  is a  $N(n+M) \times N(n+M)$  co-variance matrix;  $C_1 = \Omega I$ ;  $\Omega$  is a large positive number;  $I$  is the identity matrix.

#### IV. EXPERIMENTAL RESULTS

The proposed exTS model was tested on a well known benchmark problem and on a real data set.

##### A. Mackey-Glass time-series data set

The chaotic time series is generated from the Mackey-Glass differential delay equation defined by [11]:

$$\frac{\partial s}{\partial t} = \frac{0.2s(t-\tau)}{1+s^{10}(t-\tau)} - 0.1s(t) \quad (27)$$

The following experiment was conducted: 3000 data points, for  $t = 201 : 3200$ , are extracted from the time series and used as training data; 500 data points, for  $t = 5001 : 5500$ , are used as testing (validation) data. The learning mechanism was stopped for the testing data. The aim is to predict the value  $s(t+\Delta T)$ . The so-called non-dimensional error index (NDEI) defined as the ratio of the root mean square error (RMSE) over the standard deviation of the target data is used to compare model performance as well as the RMSE itself. The results for

TABLE I  
RESULTS FOR MACKEY-GLASS PROBLEM;  $\Delta T=85$

Algorithm	Radius	Rules	NDEI
exTS	adaptive	10	0.331
exTS, $\varepsilon$ - default	adaptive	9	0.361
eTS [11,18]	0.25, fixed	9	0.372
RAN [32]	N/A	113	0.373
ESOM [20]	N/A	114	0.32
EFuNN [21]	N/A	193	0.401
DENFIS [20]	adaptive	58	0.276

$\Delta T = 85$  are tabulated in Table I. The problem-independent parameter  $\Omega=1000$ ;  $\varepsilon=3\%$  is an optional parameter that can be used to further improve model structure. It should be noted

that in [19] the comparison with eTS has been done ([19], Table 2, p.1438), but the results included in that Table concern prediction 6 steps ahead, while the results for eTS [11] and DENFIS [20] are for 85 steps ahead, which is completely different problem.

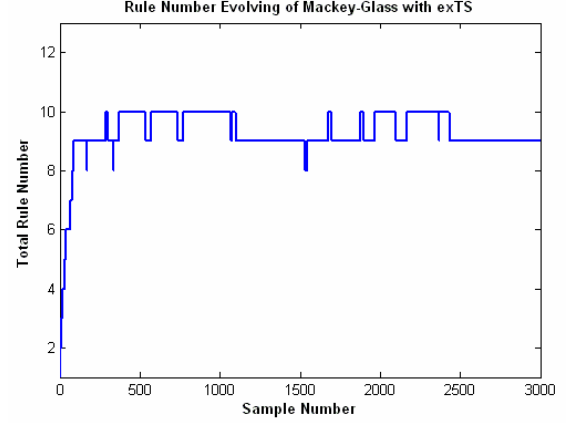


Fig. 3. Evolution of the rule-base (xTS with  $\varepsilon=3\%$ )

From the Table I it is obvious that the proposed exTS type FRB system provides better performance in terms of higher precision, lower number of fuzzy rules and lower number of parameters required.

##### B. NOx emissions in car-engine tests on-line modelling

The second test concerns real data (courtesy of Dr. E. Lughofer, Linz, Austria) generated from tests with real car engines. It concerns the problem of real-time modelling NOx emissions from the exhaust in a car engine using the following four input attributes measured in real-time:

- $N$  - engine rotation speed, *rpm*
- $P2$  - pressure offset in cylinders, *bar*
- $T$  - engine output torque, *Nm*
- $Nd$  - speed of the dynamometer, *rpm*

The non-linear dependence of the following form is expected and the proposed exTS has been used to find it based on the data alone, starting 'from scratch':

$$NOx_k = f(N_{k-4}, P2_{k-5}, T_{k-5}, Nd_{k-6}) \quad (28)$$

The sampling period of  $1,s$  has been used and a prediction  $4,s$  ahead was made. The data set consists of 1491 samples 824 of which have been used for validation only (the evolution of the xTS has been stopped). The results have been compared with the similar test published before [17] using FLEXFIS and eTS. The performance is similar, but the advantage of exTS is the more flexible way of describing the fuzzy sets which allows a better linguistic interpretation.

TABLE II  
RESULTS FOR NOX CAR ENGINE DATA SET

Algorithm	Radius	Rules	Correlation	NDEI
exTS, $\varepsilon$ - default	adaptive	5	0.916	0.4032
eTS [17]	fixed; 0.5	3	0.915	0.40567
FLEXFIS [17]	fxed	4	0.911	N/A



## V. CONCLUSION

The proposed evolving fuzzy modeling approach which concerns eXtended Takagi-Sugeno (exTS) fuzzy systems with its ability to have a higher level of *gradual* adaptation to the environment and to the changing data patterns (to *evolve*) is an efficient tool for complex modeling, prediction, and classification tasks that exist in modern process industries, finance, defense and other areas.

A significant novelty is the recursive formula for adaptation of the radius (spread of the membership functions) in a data-driven fashion. It allows flexible clusters with an evolving shape to be build that better match the data distribution. A new condition is also introduced to replace clusters in order to avoid contradictory rules at the point of entry to the rule-base. The quality of the clusters is monitored and controlled on-line by the parameters age, support, and their variable recursively adaptive radius.

The experimental results with a well known benchmark problem of Mackey Glass chaotic time series and a real data set for NO<sub>x</sub> emissions from car engines on-line modeling demonstrate the superiority of the proposed exTS FRB system in terms of precision and compatibility (thus interpretability).

Evolving FRB systems, introduced in [9]-[11] and further developed here, represent a very powerful concept that can address problems of high level adaptation of non-linear complex systems to non-stationary environment and to internal changes in these systems. Methods based on the way humans deduct knowledge from experience, this knowledge gradually evolves and is inherited are an efficient tool to cope with the enormous amount of information that is generated with fast rates. A wide range of applications of this approach is published in a number of parallel papers.

## VI. ACKNOWLEDGEMENTS

The authors would like to thank Dr. E. Lughofer for kindly providing the experimental data from the car engines, Dr. D. Filev for the very insightful comments and the overall support, as well as the anonymous referees for their useful comments.

## REFERENCES

- [1] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, *From Data Mining to Knowledge Discovery: An Overview, Advances in Knowledge Discovery and Data Mining*, MIT Press, 1996.
- [2] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. Heidelberg, Germany: Springer Verlag, 2001.
- [3] J.S.R. Jang, "ANFIS: Adaptive Network-based Fuzzy Inference Systems", *IEEE Trans. on Systems, Man & Cybernetics*, v.23 (3), pp.665-685, 1993.
- [4] T. J. Procyk, E. H. Mamdani, "A linguistic self-organizing process controller", *Automatica*, 1979, pp. 15-30.
- [5] K. Shimojima, T. Fukuda, Y. Hashegawa (1995) Self-Tuning Modeling with Adaptive Membership Function, Rules, and Hierarchical Structure based on Genetic Algorithm, *Fuzzy Sets and Systems*, v.71, pp.295-309.
- [6] P. J. Bentley, Evolving Fuzzy Detectives: An Investigation into the Evolution of Fuzzy Rules, In: *Suzuki, Roy, Ovasko, Furuhashi and Dote (Eds), Soft Computing in Industrial Applications*, Springer Verlag: London, UK, 2000.
- [7] A. S. Hornby, *Oxford Advance Learner's Dictionary*, Oxford University Press, 1974.
- [8] B. Fritze "Growing cell structures – a self-organizing network for unsupervised and supervised learning," *Neural Networks*, vol. 7 (9) pp.1441-1460, 1994.
- [9] P. Angelov, R. Buswell, "Evolving Rule-based Models: A Tool for Intelligent Adaptation", *Proc. Joint 9<sup>th</sup> IFSA World Congress and 20<sup>th</sup> NAFIPS Intern. Conf.*, Vancouver, BC, Canada, July 25-28, 2001, IEEE Press, ISBN 0-7803-7079-1, pp.1062-1066.
- [10] P. Angelov, *Evolving Rule-based Models: A Tool for Design of Flexible Adaptive Systems*. Berlin, Germany: Springer Verlag, 2002.
- [11] P. Angelov, D. Filev, "An approach to on-line identification of evolving Takagi-Sugeno models", *IEEE Trans. on Systems, Man and Cybernetics, part B*, vol.34, No1, pp. 484-498, 2004.
- [12] K. Astrom, B. Wittenmark, "Computer Controlled Systems: Theory and Design," NJ, USA: Prentice Hall, 1984.
- [13] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, Berlin, Germany: Springer Verlag, 1996.
- [14] J. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, USA: MIT Press, 1992.
- [15] P. Angelov, D. Filev, "Simpl\_eTS: A Simplified Method for Learning Evolving Takagi-Sugeno Fuzzy Models," The 2005 IEEE Intern. Conf. on Fuzzy Systems FUZZ-IEEE, Reno, NE, USA, 2005, pp.1068-1073.
- [16] P. Angelov, C. Xydeas, D. Filev, "On-line Identification of MIMO Evolving Takagi-Sugeno Fuzzy Models, Intern. Joint Conf. on Neural Networks and Intern. Conf. on Fuzzy Systems", IJCNN-FUZZ-IEEE, Budapest, Hungary, 25-29 July, 2004, pp. 55-60, ISBN 0-7803-8354-0
- [17] P. Angelov, E. Lughofer, P. E. Klement, "Two Approaches for Data-Driven Design of Evolving Fuzzy Systems: eTS and FLEXFIS," *The 2005 NAFIPS Ann. Conf.*, June 2005, Ann Arbor, MI, USA, pp.31-35.
- [18] P. Angelov, Victor, J., Dourado, A., Filev, D., "On-line evolution of Takagi-Sugeno Fuzzy Models", Proc. Of the 2nd IFAC Workshop on Advanced Fuzzy/Neural Control, 16-17 September 2004, Oulu, Finland, pp.67-72.
- [19] K. Kim, J. Baek, E. Kim, M. Park, "TSK Fuzzy model based on-line identification," *Proc. 11<sup>th</sup> IFSA World Congress*, Beijing, China, 2005, pp.1435-1439.
- [20] N. Kasabov, Q. Song "DENFIS: Dynamic Evolving Neural-Fuzzy Inference System and Its Application for Time-Series Prediction," *IEEE Trans. on Fuzzy Systems, Vol.10* (2), pp. 144-154, 2002.
- [21] N. Kasabov, "Evolving fuzzy neural networks for on-line supervised/unsupervised, knowledge-based learning," *IEEE Trans. SMC - part B, Cybernetics* 31, 902-918, 2001.
- [22] G. Leng, T. M. McGuinty, G. Prasad, "An approach for on-line extraction of fuzzy rules using a self-organizing fuzzy neural network," *Fuzzy Sets and Systems*, vol. 150 (2), 2005, pp.211-243.
- [23] F.-J. Lin, C.-H. Lin, P.-H. Shen, "Self-constructing fuzzy neural network speed controller for permanent-magnet synchronous motor drive," *IEEE Trans. on Fuzzy Systems, Vol.9* (5), pp. 751-759, 2001.
- [24] C.-F. Juang, X.-T. Lin "A recurrent self-organizing neural fuzzy inference network," *IEEE Trans. on Neural Networks*, 10, 828-845, 1999
- [25] G.-B. Huang, P. Saratchandran, N. Sundarajan, "A generalized growing and pruning RBF (GGAP-RBF) neural network for function approximation," *IEEE Trans. on NN*, vol.16 (1) 57-67, 2005.
- [26] D. Filev, "Rule-base guided adaptation for mode detection in process control", *Proc. Joint 9<sup>th</sup> IFSA World Congress and 20<sup>th</sup> NAFIPS Intern. Conf.*, Vancouver, BC, Canada, July 25-28, 2001, vol.2, pp. 1068-1073, IEEE Press, ISBN 0-7803-7079-1.
- [27] L.-X. Wang "Fuzzy Systems are Universal Approximators," *Proc. FUZZ-IEEE*, San Diego, CA, USA, pp.1163-1170, 1992.
- [28] T. Takagi, M. Sugeno, "Fuzzy identification of systems and its application to modeling and control", *IEEE Trans. on Syst., Man & Cybernetics*, vol. 15, pp. 116-132, 1985.
- [29] R. R. Yager, D. P. Filev, *Essentials of Fuzzy Modeling and Control*, New York, USA: John Wiley and Sons, 1994.
- [30] S. L. Chiu, "Fuzzy model identification based on cluster estimation," *Journal of Intelligent and Fuzzy Syst.* vol.2, pp. 267-278, 1994.
- [31] R. R. Yager, D.P. Filev, "Learning of Fuzzy Rules by Mountain Clustering," *Proc. of SPIE Conf. on Application of Fuzzy Logic Technology*, Boston, MA, USA, pp.246-254, 1993.
- [32] J. Plat, "A resource allocation network for function interpolation," *Neural Computation*, vol. 3 (2) 213-225, 1991.