

On-line Identification of MIMO Evolving Takagi-Sugeno Fuzzy Models

Plamen Angelov

Depr of Communications Syst
Lancaster University,
Lancaster, LA14YR, UK
E-mail: p.angelov@lancaster.ac.uk

Costas Xydeas

Depr of Communications Syst
Lancaster University,
Lancaster, LA14YR, UK
E-mail: c.xydeas@lancaster.ac.uk

Dimitar Filev

Ford Motor Co.
24500 Glendale av.
Detroit, MI 48239, USA
e-mail: Dfilev@Ford.com

Abstract— Evolving Takagi-Sugeno (eTS) fuzzy models and the method for their on-line identification has been recently introduced as an effective tool for design of flexible system models with minimum *a priori* information. Their structure develops *on-line* during the process of model identification itself. In this paper, this approach has been extended for the case of multi-input-multi-output (MIMO) system model. Both parts of the identification algorithm, namely the unsupervised fuzzy rule-base antecedents learning by a recursive, non-iterative clustering, and the supervised linear sub-model parameters learning by Kalman-filtering-based procedure, are extended for the MIMO case. The radius of influence of each fuzzy rule is considered a vector instead of a scalar as in the original eTS approach, allowing different areas of the data space to be covered by each input variable. As in the eTS, in MIMO eTS, the rule-base and parameters of the fuzzy model continually evolve by adding new rules with more summarization power and by modifying existing rules and parameters. Simulation results using a well-known benchmark are considered in this paper. Further investigation concern the application of MIMO eTS to predictive modeling of the speech spectrum magnitude, classification of multi-channel source modulation etc.

I. INTRODUCTION

For many practical engineering problems the so-called first principle models are difficult or impossible to build [1]. Black-box models offer a solution, but they do not give much insight into the system being modeled. Fuzzy rule-based models and especially so-called Takagi-Sugeno models have gained significant impetus during the last decade because of their computational efficiency and transparency [10].

They have a pseudo-linear nature and utilize the idea of approximation of a nonlinear system by a collection of fuzzily blended linear sub-models. The TS model representation often provides efficient and computationally attractive solutions to a wide range of modeling problems capable to approximate nonlinear dynamics, multiple

operating modes and significant parameter and structure variations [10].

In the *off-line* case *all* of the data is available at the start of the process of training and the learning involves consecutive structure and parameter identification [5,10]. Structure identification includes estimation of the focal points of the rules (antecedent parameters) by fuzzy clustering. With fixed antecedent parameters, the TS model transforms into a linear model. Parameters of the linear models associated with each of the rule antecedents are obtained by pseudo-inversion or by applying the Recursive Least Square (RLS) method [4]. Alternatively, the antecedent parameters can be considered as initial estimates only, while the structure and parameters are optimized using back-propagation [6] or genetic algorithm [9]. The original TS model [10] concerns MISO model. It has been later generalized into MIMO structure by a cascade composition [7].

The continuous *on-line* learning of TS models is based on *recursive, non-iterative* clustering method responsible for model structure (rule base) learning *on-line* and *recursive* linear model parameter estimation [3]. If, in addition, assume that the model structure is not known *a priori*, but instead it evolves *gradually* (it should be noted that this evolution is much slower than the evolution of the model parameters) then we come to the concept of eTS [1-3]. The eTS use the informative potential of the new data sample (accumulated spatial proximity measure) as a trigger to update the rule-base [1-3]. The eTS learning algorithm is robust (outliers have no chance to become rule centers because of the way of its definition). An additional mechanism for ensuring a *gradual* change of the rule-base structure and inheritance of the structural information is the replacement of a rule center by a new data sample [1-3].

It is important to note that learning could start without *a priori* information and only a single data sample. This interesting feature makes the approach potentially very useful in autonomous and smart adaptive systems [1].

The concept of eTS modeling [1] is further developed here in respect to *on-line* identification of MIMO systems. The rest of the paper is organized as follows. The problem of identification of MIMO TS models is presented in section II. Two alternative ways (globally and locally

optimal) of calculation of the consequent parameters are presented. A new vector definition of the radii of influence of the fuzzy sets is introduced. The approach for *on-line* learning MIMO eTS models is represented in section III. Section IV studies simulation results. Concluding remarks are given in section V.

II. IDENTIFICATION OF MIMO TS FUZZY MODEL

Sugeno and coworkers [10] have introduced the first systematic method for identification of fuzzy models. It concerns a special group of rule-based models with fuzzy antecedents and functional consequent, called briefly TS models that follow from the Takagi-Sugeno-Kang reasoning method:

$$\mathcal{R}^i: \text{IF } (x_1 \text{ is } \mathfrak{S}_1^i) \text{ AND } \dots \text{ AND } (x_n \text{ is } \mathfrak{S}_n^i) \\ \text{THEN } (y^i = x_e^T \pi^i) \quad i=1,2,\dots,R \quad (1)$$

where \mathcal{R}^i denotes the i^{th} fuzzy rule; R is the number of fuzzy rules; x_e^T is the extended input vector, $x_e^T = [1, x^T]$, which is formed by appending the input vector $x = [x_1, x_2, \dots, x_n]^T$ with 1 (allowing a free parameter); \mathfrak{S}_j^i denotes the antecedent fuzzy sets, $j=1,2,\dots,n$; y^i is the output of the i^{th} linear subsystem; $\pi^i = [a_0^i, a_1^i, \dots, a_m^i]^T$ are its parameters.

In this paper, we consider a MIMO extension of the TS model, where the output is multidimensional. This results in the set of equations similar to (1) with the important difference in the variable definition, namely y^i now denotes the multidimensional **vector** output of the i^{th} linear subsystem, $y^i = [y_1^i, y_2^i, \dots, y_m^i]^T$;

$$\pi^i = \begin{bmatrix} a_{01}^i & a_{02}^i & \dots & a_{0m}^i \\ a_{11}^i & a_{12}^i & \dots & a_{1m}^i \\ \dots & \dots & \dots & \dots \\ a_{n1}^i & a_{n2}^i & \dots & a_{nm}^i \end{bmatrix} \text{ are its parameters.}$$

At the heart of the TS method for fuzzy modeling is the segmentation of the data space into fuzzily defined regions. The fuzzy regions are parameterized and each region is associated with a linear sub-system. As a result, the nonlinear system forms a collection of loosely (fuzzily) coupled (blended) multiple linear models. The degree of firing of each rule is proportional to the level of contribution of the corresponding linear model to the overall output of the TS model. Without loss of generality we use Gaussian antecedent fuzzy sets, which ensures greatest possible generalization of the description:

$$\mu^i(x) = e^{-4 \left\| \frac{x - x^{i*}}{r} \right\|^2}; i=1,2,\dots,R \quad (2)$$

where r is a positive constant, which defines the radius of the antecedent and the zone of influence of the i^{th} model; x^{i*} is the focal point of the i^{th} rule antecedent.

The radius r is one of the few parameters of the algorithm, which needs to be pre-defined. Its value is an important leverage for a trade-off between the model complexity and precision [8]. As a general guidance, too large values of r lead to averaging, too small - to over-fitting. A value of r in the range of [0.3; 0.5] has been recommended [1-3,8]. Since the parameters of the linear models are boundless, there is no need for normalization of the inputs. However, when the values of inputs differ significantly a vector representation of the radius gives more flexibility and can compensate the weights of the projections of the distance between a data point and a rule center on different inputs:

$$r = [r_1, r_2, \dots, r_n]^T \quad (3)$$

The vector of radii can be expressed as a proportion of the expected range of each variable:

$$r = \bar{r}(\bar{x} - \underline{x}) \quad (3a)$$

where $\bar{x} = [\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n]^T$ is the vector of expected maximums of the inputs; $\underline{x} = [\underline{x}_1, \underline{x}_2, \dots, \underline{x}_n]^T$ is the vector of expected minimums of the inputs; the recommended value for \bar{r} is [0.3; 0.5].

It should be noted that in *on-line* mode one could only expect the range of each of the inputs, but the precise values of \bar{x} and \underline{x} are not critically important as they are compensated to some extent by the value of \bar{r} , and ultimately by the boundless consequent parameters.

In this paper we use different radii for each of the input variables:

$$\mu^i(x) = e^{-4 \sum_{j=1}^n \frac{(x_j - x_j^{i*})^2}{r_j^2}} \quad (2a)$$

The TS model output is calculated by weighted averaging of individual rules' contributions:

$$y = \sum_{i=1}^R \lambda^i(x) y^i = \sum_{i=1}^R \lambda^i(x) x_e^T \pi^i \quad (4)$$

where $\lambda^i(x) = \frac{\mu^i(x)}{\sum_{j=1}^R \mu^j(x)}$ is the normalized firing level

of the i^{th} rule;

It can be represented in a vector form as:

$$y = \psi^T \theta \quad (4a)$$

where $\theta = [(\pi^1)^T, (\pi^2)^T, \dots, (\pi^R)^T]^T$ is a vector composed of the linear model parameters;

$\psi = [\lambda^1 x_e^T, \lambda^2 x_e^T, \dots, \lambda^R x_e^T]^T$ is a vector of the inputs that are weighted by the normalized firing levels of the rules.

For a given set of input-output data (x_k^T, y_k) , $k=[1, TD]$, TD is the number of training data samples (in *off-line* mode they are fixed) and fixed antecedent parameters the vector of sub-model parameters θ should minimize the following objective function:

$$J_G = \sum_{k=1}^{TD} (y_k - \psi_k^T \theta)^2 \quad (5)$$

The optimal solution can be estimated by the RLS algorithm called also the Kalman filter [4,10]. The objective function (5) is **globally** optimal, but this does not guarantee locally adequate behavior of the sub-models that form the TS model [3,11]. Locally meaningful sub-models could be found using the locally weighted objective function [3]:

$$J_L = \sum_{i=1}^R (Y - X^T \pi_i)^T \Lambda_i (Y - X^T \pi_i) \quad (6)$$

A set of solutions to individual cost functions J_{Li} associated with *each* rule can be recursively calculated through the weighted RLS (wRLS) algorithm [3].

III. ON-LINE LEARNING OF MIMO ETS MODELS

In *on-line* mode, the training data are collected continuously, rather than being a fixed set. *On-line* learning of eTS models includes *recursive* clustering under assumption of a *gradual* change of the rule-base and wRLS method [3].

The *on-line* clustering procedure starts (**stage 1**) with the first data point established as the focal point of the first cluster ($i=1$). Its coordinates are used to form the antecedent part of the fuzzy rule (1) using for example Gaussian membership functions (2). Its potential, P is assumed equal to 1.

$$k:=1; R:=1; x^* := x_k; P_1(z^*) := 1; \theta_1 := \pi_1 = 0; C_1 = \Omega \quad (7)$$

where z^{1*} is the first cluster center; x^{1*} is focal point of the first rule being a projection of z^{1*} on the axis x .

Starting from the next data point ($k:=k+1$) onwards (**stage 2** in a loop) the potential of the new data points (z_k) is calculated *recursively*. As a measure of the potential, we use a Cauchy type function of first order:

$$P_k(z_k) = \frac{1}{1 + \frac{1}{(k-1) \sum_{i=1}^{k-1} \sum_{j=1}^{n+m} (d_{ik}^j)^2}}; k=2,3,\dots \quad (8)$$

where $P_k(z_k)$ denotes the potential of the data point (z_k) calculated at time k ; $d_{ik}^j = z_i^j - z_k^j$, denotes projection of

the distance between two data points (z_i^j and z_k^j) on the axis z^j (x^j for $j=1,2,\dots,n$ and on the axis y for $j=n+m$).

It should be noted that since we consider MIMO system each data sample is $(n+m)$ dimensional.

Potential of the *new* data sample is *recursively* calculated at **stage 3** in the same loop as follows:

$$P_k(z_k) = \frac{k-1}{(k-1)(\mathcal{G}_k + 1) + \sigma_k - 2\nu_k} \quad (9)$$

where $\mathcal{G}_k = \sum_{j=1}^{n+m} (z_k^j)^2$; $\sigma_k = \sum_{l=1}^{k-1} \sum_{j=1}^{n+m} (z_l^j)^2$; $\nu_k = \sum_{j=1}^{n+m} z_k^j \beta_k^j$; $\beta_k^j = \sum_{l=1}^{k-1} z_l^j$.

Parameters \mathcal{G}_k and ν_k in (9) are calculated from the current data point z_k , while β_k^j and σ_k are recursively updated as follows:

$$\sigma_k = \sigma_{k-1} + \sum_{j=1}^{n+m} (z_{k-1}^j)^2; \beta_k^j = \beta_{k-1}^j + z_{k-1}^j.$$

After the *new* data are available in *on-line* mode, they influence the potentials of the centers of the clusters (z^{l*} , $l=1,2,\dots,R$), which are respective to the focal points of the existing rules (x^{l*} , $l=1,2,\dots,R$). The reason is that by definition the potential depends on the distance to *all* data points, including the new ones (the sum in the denominator by i in (8) has an increasing number of components).

At **stage 4** in the same loop the potentials of the focal points of the existing clusters are updated *recursively*:

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

where $P_k(z^{l*})$ is the potential of the l -th cluster z^{l*} , which is a prototype of the l -th rule at time k .

At **stage 5** in the loop the potential of the *new* data point is compared to the updated potential of the centers of the existing clusters and a decision whether to modify or upgrade the rule-base is taken. The evolution of the rule-base is driven by the following two basic principles:

Principle 1 (REPLACE):

IF the potential of the new data point (z_k) is higher than the potential of the existing rule centers:

$$P_k(z_k) > \max_{i=1}^R P_k(z^{i*}) \quad (11)$$

AND z_k is close to an existing rule center given by:

$$\frac{P_k(z_k)}{\max_{i=1}^R P_k(z^{i*})} - \frac{\delta_{\min}}{\|r\|} > 1 \quad (12)$$

where $\delta_{\min} = \min_{i=1}^R \|z_k - z^{i*}\|$ denotes the distance from the new data point to the closest of the existing rule centers;

$\|r\| = \sqrt{\frac{1}{n} \sum_{i=1}^n r_i^2}$ is the norm of the radii.

THEN the new data point (z_k) **replaces this center** (let us suppose that it has index j):

$$x^{j*} := x_k; P_k(z^{j*}) := P_k(z_k) \quad (13)$$

Principle 2 (ADD):

IF only (11) is satisfied but not (12)

THEN the new data point is **added** to the rule-base as a new center and a new rule is formed with a focal point based on the projection of this center on the axis x ($R := R+1; x^{R*} = x_k$).

$$R := R+1; x^{R*} = x_k; P_k(z^{R*}) = P_k(z_k) \quad (14)$$

In eTS the rule-base *gradually evolves* [1-3]. Therefore the normalized firing strengths of the rules (λ^i) change, which affects *all* the data (including the data collected before time of the change). Therefore, the straightforward application of the RLS or wRLS is not correct [3]. A resetting of the covariance matrices and parameters initialization of the RLS is made at each time a new $(R+1)^{th}$ rule is added to the rule base estimating them as a weighted average of the respective covariance and parameters of the remaining R rules [3].

In case when the globally optimal objective function is minimized (5) the following RLS procedure is applied (**stage 6** in the loop):

$$\hat{\theta}_k = \hat{\theta}_{k-1} + C_k \psi_{k-1} (y_k - \psi_{k-1}^T \hat{\theta}_{k-1}) \quad k=2,3,\dots \quad (15)$$

$$C_k = C_{k-1} - \frac{C_{k-1} \psi_{k-1} \psi_{k-1}^T C_{k-1}}{1 + \psi_{k-1}^T C_{k-1} \psi_{k-1}} \quad (16)$$

where $\hat{\theta}_1 = \begin{bmatrix} (\hat{\pi}^1)^T, (\hat{\pi}^2)^T, \dots, (\hat{\pi}^R)^T \end{bmatrix}^T = 0; C_1 = \Omega I$.

When a new rule is added (ADD) to the rule-base, the RLS is reset in the following way [3]:

i) Parameters of the new rule are determined by the weighted average of the parameters of the other rules. The weights are the normalized firing levels of the existing rules λ^i . Parameters of the other rules are *inherited* from the previous step [3]:

$$\hat{\theta}_k = \begin{bmatrix} (\hat{\pi}_{k-1}^1)^T, (\hat{\pi}_{k-1}^2)^T, \dots, (\hat{\pi}_{k-1}^R)^T, (\hat{\pi}_{k-1}^{R+1})^T \end{bmatrix}^T \quad (17)$$

$$\text{where } \pi_k^i = \sum_{j=1}^{R+1} \lambda^j \pi_{k-1}^{j,i} \quad (18)$$

ii) Co- variance matrices are reset as:

$$C_k = \begin{bmatrix} \rho \zeta_{11} & \dots & \rho \zeta_{1R(n+1)} & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \rho \zeta_{R(n+1)1} & \dots & \rho \zeta_{R(n+1)R(n+1)} & 0 & \dots & 0 \\ 0 & 0 & 0 & \Omega & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & \Omega \end{bmatrix} \quad (19)$$

where $\zeta_{ij}, i=1,2,\dots,R(n+1); j=1,2,\dots,R(n+1)$ is an element of the covariance matrix; $\rho = \frac{R^2+1}{R^2}$ [3].

When a rule is replaced (REPLACE) with another one the covariance matrices are inherited from the previous time step.

Finally, in the same loop at **stage 7** having the estimated parameters of the linear sub-models we can predict the next value of the outputs in *on-line*:

$$\hat{y}_{k+1} = \psi_k^T \hat{\theta}_k \quad k=2,3,\dots \quad (20)$$

In case the locally optimal objective function (8) is minimized, the local parameter estimation is based on wRLS (**stage 6** in the loop):

$$\hat{\pi}_k^i = \hat{\pi}_{k-1}^i + c_k^i x_{ek-1} \lambda^i(x_{k-1}) (y_k - x_{ek-1}^T \hat{\pi}_{k-1}^i) \quad k=2,3,\dots \quad (21)$$

$$c_k^i = c_{k-1}^i - \frac{\lambda^i(x_{k-1}) c_{k-1}^i x_{ek-1} x_{ek-1}^T c_{k-1}^i}{1 + \lambda^i(x_{k-1}) x_{ek-1}^T c_{k-1}^i x_{ek-1}} \quad (22)$$

with initial conditions $\pi_1^i = 0; c_1^i = \Omega I; i=1,2,\dots,R$

In this case, the covariance matrices are separate for each rule and have smaller dimensions ($c_k^i \in R^{(n+1) \times (n+1)}; i=1,2,\dots,R$). Parameters of the newly added rule (when ADD is activated) are determined as weighted average of the parameters of the rest R rules by (18). Parameters of the other R rules are inherited ($\pi_k^i := \pi_{k-1}^i; i=1,2,\dots,R$). When a rule is replaced by another rule (when REPLACE is activated) then parameters of all rules are inherited ($\pi_k^i := \pi_{k-1}^i; i=1,2,\dots,R$). The covariance matrix of the newly added rule (ADD) is initialized by:

$$C_k^{R+1} = \Omega I \quad (23)$$

The covariance matrices of the rest R rules are inherited ($c_k^i := c_{k-1}^i; i=1,2,\dots,R$).

The *recursive* procedure for *on-line* learning of MIMO eTS models includes the following stages:

Stage 1: Initialization of the rule-base structure.

Stage 2: Reading the *next* data sample.

Stage 3: **Recursive** calculation of the potential of each new data sample.

Stage 4: **Recursive** up-date of the potentials of existing centers.

Stage 5: Comparison of the potentials of the new data sample and the existing centers. Evolution of the

rule-based structure based on the closeness of the new data point to the existing focal points.

Stage 6: **Recursive** calculation of the consequent parameters.

Stage 7: Prediction of the outputs for the next time step using the MIMO eTS model.

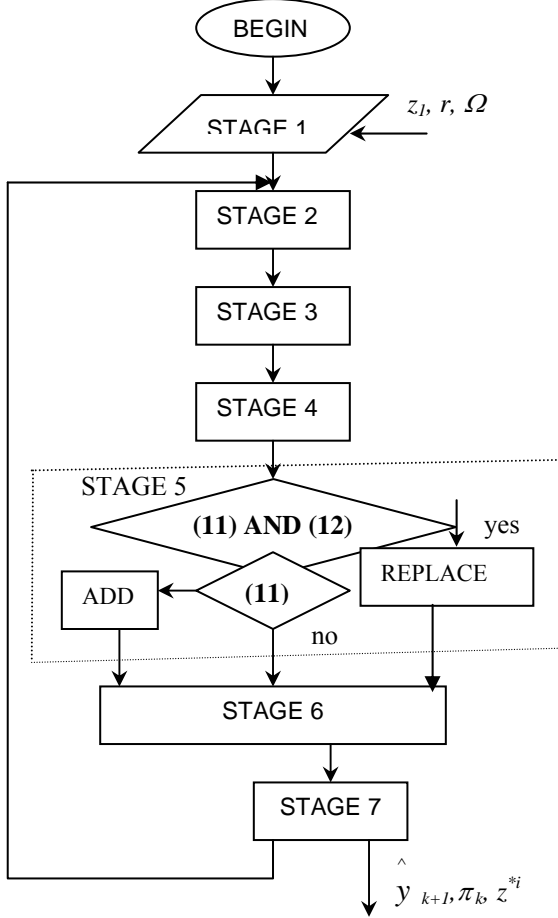


Fig. 1 Flow chart of the algorithm of eTS learning

IV. EXPERIMENTAL RESULTS

The proposed approach has been tested on a benchmark problem: prediction of the Mackey-Glass chaotic time series, which is generated from the differential delay equation defined by [2,3,5]:

$$\dot{v}(t) = \frac{0.2v(t-17)}{1+v^{10}(t-17)} - 0.1v(t); \quad v(0)=1.2$$

The aim is using the past values of v to predict future values of v . The value of the signal 6, 12 and 85 steps ahead $y=[v(t+6), v(t+12), v(t+85)]$ is predicted based on the values of the signal at the current moment, 6,12, and 18 steps back: $x=[v(t-18), v(t-12), v(t-6), v(t)]^T$. The only parameters pre-specified in the algorithm are $r=0.4$ (the same is used in both [3] and [8]) and $\Omega=10^9$.

The training set consists of the first 300 data samples only and the validation set consists of the next 500 data samples. It should be noted that the learning mechanism is **not** active for the validation unlike [8]. During validation the rule set consists of the 11 rules generated in the training phase from scratch based on the data only and no *a priori* information. Moreover, the parameters of the linear sub-models (consequent) are also fixed (the Kalman filters are not active). The non-dimensional error index defined as the ratio of the root mean square error over the standard deviation of the target data is used to evaluate the model performance in both training and validation phase (NDEIt, RMSEt; NDEIv, NDEIv).

The results (Table 1, Figures 2 and 3) illustrate that the MIMO eTS can efficiently predict a vector of outputs (in this case the value of the chaotic signal 6, 12, and 85 steps ahead simultaneously) using a transparent fuzzy rule-based model.

TABLE 1 PREDICTION ERROR AND NUMBER OF RULES (MIMO ETS - FIRST 3 ROWS; ETS [3] - NEXT ROW (SAME - SCENARIO C [8]; DIFFERENT SCENARIA OF ETS [8] REMAINING 4 ROWS)

Error	RMSEt	NDEIt	RMSEv	NDEIv	R
$y_1=v(t+6)$	0.0771	0.3428	0.0726	0.3203	11
$y_2=v(t+12)$	0.0925	0.4114	0.0893	0.3941	11
$y_3=v(t+85)$	0.1015	0.4479	0.1039	0.4547	11
[3]/[8],C	0.0855	0.3775	0.0867	0.3827	19
[8], D	0.0962	0.4250	0.0939	0.4146	6
[8], E	0.0827	0.3654	0.0802	0.3542	15
[8], F	0.0726	0.3208	0.0721	0.3183	17
[8], G	0.0865	0.3820	0.0848	0.3744	11

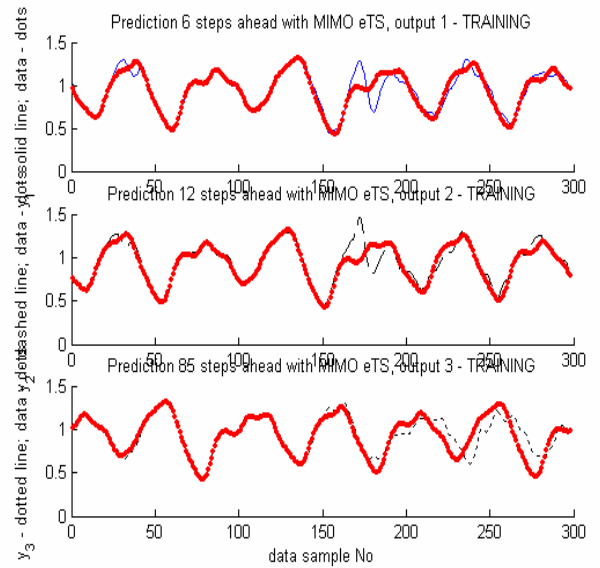


Fig. 2 MIMO eTS model training in prediction 6, 12, and 85 steps ahead. Mackey Glass chaotic time series - dots; MIMO eTS output 1 - solid line on the top plot; output 2 - dashed line on the middle plot; output 3 - dotted line on the bottom plot.

The fuzzy rule-based model has evolved to 11 rules formed around data samples with indices {1; 42;43;44;45;89;93;94;95;96;97} during the very short training phase (300 samples, while in [3] and [8] 3000 training samples are used. It should also be noted that during the validation the rule base and the sub-model parameters have been fixed unlike [8]. Since the identification is on-line even during the training phase the model is used for prediction of unseen data sample, therefore the experimental set-up used in [8] is perfectly fair. The process of model evolution, however, is not stopped there, which requires on-line supply of training samples for the output, which is not always feasible. Therefore, here we have adopted a more harsh validation set-up, which is the same as in the off-line model training - exposure of the trained model to unseen samples with fixing its structure and parameters. Therefore, the results in Table 1 are less favorable for 85 steps ahead prediction (the same used in [3] and [8]) for the MIMO eTS, though they are better for the other two predicted cases. But notably they are of the same order of magnitude. The number of rules generated using MIMO model is similar to the number of rules generated using eTS [3], [8]. It should be noted that in MIMO eTS the same set of rules is used to predict three different outputs and the data space dimension is higher.

V. CONCLUDING REMARKS

The concept of evolving Takagi-Sugeno models (eTS) has been further developed in this paper for the case of MIMO systems. The radii of influence of the fuzzy sets are defined as a vector, which gives more flexibility in the problem definition. Using the approach, a transparent, compact and accurate model can be found by rule base evolution based on experimental data with the simultaneous *recursive* estimation of the fuzzy set parameters. It should be noted that all steps of the algorithm are *non-iterative*.

The main advantages of the approach are:

- ✓ it can develop/evolve an existing model when the data pattern changes;
- ✓ it can start to learn a process from a single data samples and improve the performance of the model predictions *on-line*;
- ✓ it is *non-iterative* and *recursive* and hence computationally very effective.

The results illustrate the viability, efficiency and the potential of the approach when used with a limited amount of initial information, especially important in autonomous systems and robotics. Further investigation concern the application of MIMO eTS to predictive modeling of the speech spectrum magnitude, classification of multi-channel source modulation etc.

REFERENCES

- [1] Angelov P. P., "Evolving rule-based models: a tool for design of flexible adaptive systems", Springer-Verlag, Heidelberg, 2002.
- [2] Angelov, P., R Buswell, "Identification of evolving fuzzy rule-based models", *IEEE Transactions on Fuzzy Systems*, vol.10, No.5, 2002, pp.656-667
- [3] Angelov, P., D. Filev, "An approach to on-line identification of evolving Takagi-Sugeno models", *IEEE Trans. on Systems, Man and Cybernetics, part B*, vol.34, No.1, 2004, 484-498
- [4] Åstroem, K. J., B. Wittenmark, Adaptive Control, Addison Wesley, MA, USA, 1989.
- [5] Chiu, S. L., "Fuzzy model identification based on cluster estimation", *J. of Intel. & Fuzzy Syst.* vol.2, 267-278, 1994.
- [6] Lin, F.-J., 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, no.5, pp.751-759, 2001.
- [7] Stanimir Mollov, Robert Babuska, Henk B. Verbruggen, "Analysis of Interactions in MIMO Takagi-Sugeno Fuzzy Models". In Proc. FUZZ-IEEE 2001: 769-773
- [8] Ramos, J.-V., P. Angelov, A. Dourado, Rule-Base On-line Development by Real-time Clustering, World Automation Congress, 2004, to appear
- [9] Shimojima, K., T. Fukuda, Y. Hasegawa, "Self-tuning fuzzy modeling with adaptive membership function, rules, and hierarchical structure based on genetic algorithm, *Fuzzy Sets and Syst.*, vol.71, pp.295-309, 1995.
- [10] Takagi, T., 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.
- [11] Yen, J., L. Wang, C. W. Gillespie, "Improving the Interpretability of TSK Fuzzy Models by Combining Global and Local Learning", *IEEE Trans. on Fuzzy Syst.* vol. 6, pp. 530-537, 1998

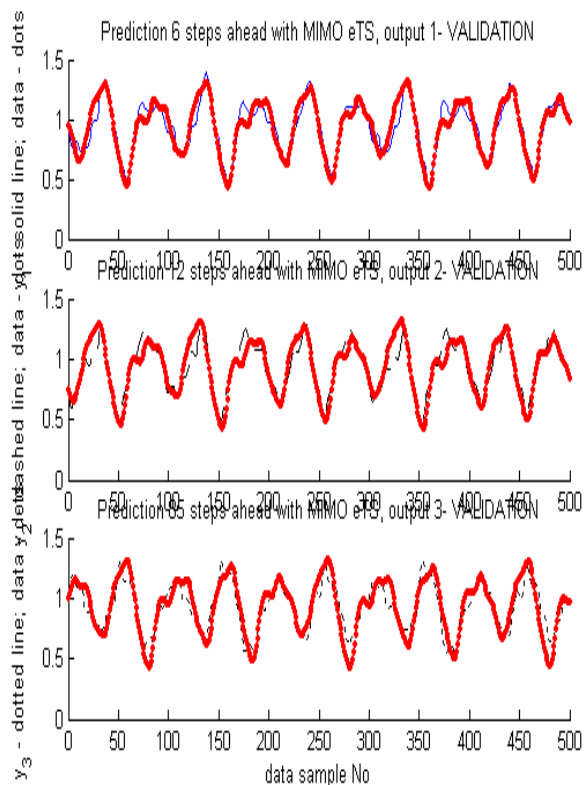


Fig. 3 MIMO eTS model validation in prediction 6, 12, and 85 steps ahead. Same symbols as in the above figure.