A New High-order Takagi-Sugeno Fuzzy Model Based on Deformed Linear Models

Ahmad Kalhor [*i], Babak N. Araabi [i] and Caro Lucas[i]

Received 14 November 2010; received in revised 5 December 2010; accepted 11 December 2010

ABSTRACT

Amongst possible choices for identifying complicated processes for prediction, simulation, and approximation applications, high-order Takagi-Sugeno (TS) fuzzy models are fitting tools. Although they can construct models with rather high complexity, they are not as interpretable as first-order TS fuzzy models. In this paper, we first propose to use Deformed Linear Models (DLMs) in consequence parts of a TS fuzzy model, which provides both complexity and interpretability. We then prove that in order to minimize considered error indices, linear and nonlinear parts of DLMs can be optimized independently. A localization of DLMs in input-space of the TS fuzzy model is done using an appropriate sigmoid-based membership function, which can represent a fuzzy subspace with enough smoothness and flat top. An incremental algorithm is also proposed to identify the suggested fuzzy model. Then, through an illustrative example, the formation of DLMs to approximate a nonlinear function is demonstrated. The applicability and effectiveness of the introduced fuzzy modeling approach is examined in three case studies: prediction of a chaotic time series, identification of a steam generator model, and approximation of a nonlinear function for a sun sensor. The obtained results demonstrate the higher accuracy and better generalization of our modeling approach as compared with those of some other well-known state-of-the-art approaches.

KEYWORDS

Takagi Sugeno fuzzy model; identification; deformed linear models; sigmoid-based membership function; prediction and approximation.

1. INTRODUCTION

Parametric nonlinear models are a good option for identification of nonlinear behaviors of a process. Alongside classical architectures and artificial neural networks, fuzzy systems introduce a vast useful family of nonlinear models. Takagi-Sugeno (TS) fuzzy systems are one of the most useful classes of fuzzy systems [1]. Consider a TS fuzzy model with fuzzy rules; the fuzzy rule is described as follows:

$$\begin{aligned}&\text{IF } (x_1 \text{ is } \tau_1^k) \text{ and } (x_2 \text{ is } \tau_2^k) \text{ and} \ldots (x_n \text{ is } \tau_n^k)\\ &\qquad \text{THEN } \tilde{y}^k = f_k(\mathbf{x}) \qquad k = 1,2,\ldots,M\end{aligned} \tag{1}$$

where $\mathbf{x}^T = [x_1\, x_2 \ldots x_n]$ denotes an $n-$dimensional input, τ_j^k is $k-$th fuzzy set for x_j, $j = 1,2,\ldots,n$ and $\tilde{y}^k = f_k(\cdot)$ is a direct mapping of input-space for the consequent part of $k-\text{th}$ fuzzy rule. If a multiplication operator is used for T-norm, the $k-\text{th}$ rule Membership Function (MF) will be $\kappa_k(\mathbf{x}) = \prod_{k=1}^{M} \tau_j^k(x_j)$. The overall output $\hat{y}$ is computed as a weighted mean value over all M rules according to:

$$\hat{y} = \sum_{k=1}^{M} \psi_k(\mathbf{x}) \tilde{y}^k(\mathbf{x}) \quad \psi_k(\mathbf{x}) = \kappa_k(\mathbf{x}) / \sum_{h=1}^{M} \kappa_h(\mathbf{x}) \tag{2}$$

From a different perspective –which we will consider in the rest of the paper, a TS fuzzy model consists of sub-models (consequent parts), which are localized in fuzzy subspaces by MFs. With respect to orders of functions $f_k(\mathbf{x})$, we have zero-order, first-order or high-order TS fuzzy models.

To localize sub-models in appropriate fuzzy subspaces and concurrently supply required nonlinearities, each identification approach must posses enough complexity.

* Corresponding Author
i Authors are with the Control and Intelligent Processing Center of Excellence, School of Electrical and Computer Engineering, University of Tehran, P.O. Box 14395/515, Tehran 14395, Iran. tel: 0098-21-82084904, fax: 0098-21-88778690, (emails: akalhor@ut.ac.ir, araabi@ut.ac.ir, lucas@ ut.ac.ir)
Babak N. Araabi and Caro Lucas are also with the School of Cognitive Sciences, Institute for Research in Fundamental Sciences (IPM), Tehran, Iran.

Accordingly, numerous complex architectures for TS fuzzy models and their corresponding learning algorithms have been developed [2]-[22].

Parameter adaptation techniques are one of alternatives to provide the required complexity. Adaptive Network-based Fuzzy Inference System (ANFIS) [2] is a known instance, where parameters, in a five layer network structure, are adapted through back-propagation and may also be combined with a least squares estimation. In [3], the Dynamic Evolving Neural-Fuzzy Inference System (DENFIS) with adaptive online and offline learning algorithms is proposed. In [4], an evolving approach to identify a hierarchical structure for a TS fuzzy model is presented, where structural improvement and parameter tuning are performed sequentially. Applying fuzzy clustering methods for identification of suitable fuzzy sets is another acceptable alternative to achieve the required complexity in a fuzzy model. Modified versions of Gath-Geva clustering have been developed in [5] and [6] to identify the premise parts of the fuzzy rules. In [7], an identification approach through hybridization of fuzzy clustering and support vector learning is suggested for a Takagi-Sugeno(TS)-type fuzzy network. In [8], an online fuzzy clustering is introduced which is independent of common spatial features. This clustering approach is utilized in identification of a developed version of TS model. In [9], an improved version of fuzzy c-regression model clustering (NFCRMA) is introduced for identification of a TS fuzzy model. In this approach, orthogonal least square is exploited to identify the consequent part parameters. Evolutionary algorithms, such as genetic algorithms, provide global optimization approaches for identification of better structure and appropriate inputs for TS fuzzy models. [10], [11] and [12] are examples of this. In some approaches, using recurrent structures in a TS fuzzy model provides the required complexity, [13][14][15]. In some other approaches, using a sequence of structural optimization and parameter tuning is proposed. In [16], a model-tree learning algorithm is proposed. There, at each step, a rule is selected and split into two new ones. In a method extending this, merging of fuzzy rules is suggested in addition to splitting, [17] and [18]. In [19], to identify a first order TS fuzzy model, the structure optimization and parameter tuning are performed sequentially, based on an output error index.

In another set of alternatives, there exist several identification approaches for high-order TS fuzzy models. In [3], a type of DENFIS that utilizes a set of Multi-Layer Perceptron(MLP) network as nonlinear sub-models in TS fuzzy model is used. Parameters of these sub-models are trained with the back-propagation (BP) algorithm. In [20], to cancel noise of a temporal process, a high-order recurrent neuro-fuzzy system is suggested. Actually, in this recurrent model, the feedback connections are implemented through finite impulse response synaptic filters leading to a higher-order network with enhanced temporal capabilities. In [21], a specified class of artificial neural networks, Extreme Learning Machines(ELM), are suggested to be integrated with fuzzy logic. The result is a high-order Takagi-Sugeno-Kang fuzzy model where sub-models are derived from multiple ELMs. In [22], quadratic polynomials have been used as sub-models, which are optimized through a version of parallel genetic algorithms. High-order TS fuzzy models, unlike first-order ones, can supply required nonlinearities mainly through nonlinear sub-models. However, optimization of their nonlinear sub-models is more challenging and the resulting fuzzy model is not interpretable like first-order TS fuzzy models.

In this paper, we suggest specific nonlinear sub-models which can be optimized partially and can be interpreted like linear sub-models. Each sub-model is the product of a linear model and a quadratic function. Through two theorems it is, then, proven that linear and nonlinear parts of the sub-models can be optimized independently, locally and globally. It is explained afterwards, how we can interpret the resulting high-order TS fuzzy model like first-order ones. We call such sub-models "Deformed Linear Models" (DLMs) and the resulting fuzzy model as a TS-DLM fuzzy model.

The rest of the paper is organized as follows: Section 2 explains about DLMs and their optimization in the TS-DLM fuzzy model. In Section 3, we introduce sigmoid MFs to localize DLMs in input-space. Section 4 explains about an incremental identification approach for TS-DLM. Section 5 is devoted to case studies: A function approximation problem is considered, which illustrates how DLMs are localized in input-space. Next, the TS-DLM fuzzy model is evaluated and compared in prediction of a chaotic time series, identification of a steam generator model, and approximation of a nonlinear function for a sun sensor. Finally, the paper is concluded in Section 6.

2. DEFORMED LINEAR MODELS

Consider a linear model $\bar{y} = \mathbf{z}^T \boldsymbol{\theta}_k$, where $\bar{y} \in \mathbf{R}$, $\mathbf{x} \in \mathbf{R}^{n\times 1}$ and $\boldsymbol{\theta} \in \mathbf{R}^{(n+1)\times 1}$ denote its output, input and linear parameters, respectively; and also, we have $\mathbf{z}^T = [\mathbf{x}^T\, 1]$. Corresponding to the considered linear model, a DLM is defined as follows:

$$\tilde{y} = \bar{y}\beta(\mathbf{x}) \qquad \beta(\mathbf{x}) = 1 - (\mathbf{x}-\mathbf{c})^T \mathbf{S}(\mathbf{x}-\mathbf{c}) \tag{3}$$

The quadratic function $\beta(\mathbf{x})$ is a nonlinear coefficient for the linear model; $\mathbf{S} \in \mathbf{R}^{n\times n}$ is a symmetric matrix and $\mathbf{c} \in \mathbf{R}^{n\times 1}$ is a focal point for $\beta(\mathbf{x})$. The coefficient at focal point is equal to one; however, it can be convexly increased or decreased in different directions with respect to eigenvectors and eigenvalues of $\mathbf{S}$. Fig. 1 shows a given 2-dimensional linear model and some of its generated DLMs through Equation (3).

In this work, we utilize some instances of the defined DLM in Equation (2) as sub-models in a TS fuzzy model as follows:

$$\hat{y}=\sum_{k=1}^{M}\psi_k(\mathbf{x})\tilde{y}^k \qquad \tilde{y}^k=\bar{y}^k\beta_k(\mathbf{x})$$
$$\bar{y}^k=\mathbf{z}^T\boldsymbol{\theta}_k, \qquad \beta_k(\mathbf{x})=(1-(\mathbf{x}-\mathbf{c}_k)^T\mathbf{S}_k(\mathbf{x}-\mathbf{c}_k)) \tag{4}$$

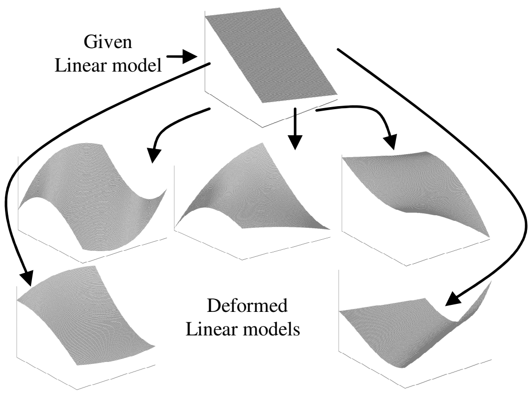

Figure 1: Equation (3) is used to created several DLMs that are derived from a given linear model

One can easily interpret the above defined high-order TS fuzzy model like the first-order ones: It is sufficient to consider $\beta_k(\mathbf{x})\psi_k(\mathbf{x})$ as a non-normalized MF and leave linear parts of DLMs as linear sub-models in TS fuzzy model. This new interpretation of TS-DLM fuzzy model facilitates its development in some uses like control applications. In the following theorems, it is proven that DLMs in the TS-DLM fuzzy model can be optimized partially globally and locally. We use '$\otimes$' notation for Kronecker product and '$vec(.)$' as an operator to vectorize a matrix [23].

Theorem 1: *Consider a nonlinear process with a single output, $y\in\mathbf{R}$, and an n-dimensional input, $\mathbf{x}\in\mathbf{R}^{n\times1}$; we denote observed input-output data points of the process with $(\mathbf{x}_i,y_i), i=1,2,\ldots,Q$; Consider the represented TS-DLM fuzzy model in Equation(4) for this process. It can be shown that:*

a) Suppose $\boldsymbol{\theta}_k, k=1,2,\ldots,M$ are known. $\mathbf{S}_k\in\mathbf{R}^{n\times n}$, $k=1,2,\ldots,M$ which satisfy Equation (5), are optimum solutions to minimizing $J=\sum_{i=1}^{Q}(y_i-\hat{y}_i)^2$.

$$\begin{bmatrix}\mathbf{B}_{11} & \mathbf{B}_{12} & \cdots & \mathbf{B}_{1M}\\ \mathbf{B}_{21} & \mathbf{B}_{22} & \cdots & \mathbf{B}_{2M}\\ \vdots & \vdots & \ddots & \vdots\\ \mathbf{B}_{M1} & \mathbf{B}_{M2} & \ldots & \mathbf{B}_{MM}\end{bmatrix}\begin{bmatrix}vec(\mathbf{S}_1)\\ vec(\mathbf{S}_2)\\ \vdots\\ vec(\mathbf{S}_M)\end{bmatrix}=\begin{bmatrix}\mathbf{C}_1\\ \mathbf{C}_2\\ \vdots\\ \mathbf{C}_M\end{bmatrix} \tag{5}$$
$$\mathbf{B}_{fk}=\sum_{i=1}^{Q}\bar{y}_i^k\bar{y}_i^f(\mathbf{A}_i^{kf}\otimes\mathbf{A}_i^{kf}),\ f=1,2,\ldots,M$$
$$\mathbf{C}_f=-vec(\sum_{i=1}^{Q}\mathbf{A}_i^{ff}\bar{y}_i^f\bar{e}_i)$$

where $\mathbf{A}_i^{fk}=(\mathbf{x}_i-\mathbf{c}_f)(\mathbf{x}_i-\mathbf{c}_k)^T$, $\bar{y}_i^f=\psi_f(\mathbf{x}_i)\mathbf{z}_i^T\boldsymbol{\theta}_f$ *and* $\bar{e}_i=y_i-\sum_{k=1}^{M}\psi_k(\mathbf{x}_i)\mathbf{z}_i^T\boldsymbol{\theta}_k$.

b) Suppose $\mathbf{S}_k, k=1,2,\ldots,M$ are known. The computed $\boldsymbol{\theta}_k, k=1,2,\ldots,M$ through Equation (6) provide optimum solutions to minimizing J.

$$\begin{bmatrix}\boldsymbol{\theta}_1^T & \boldsymbol{\theta}_2^T & \ldots & \boldsymbol{\theta}_M^T\end{bmatrix}=\zeta^T$$
$$\zeta=(\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{Z}^T\mathbf{Y} \tag{6}$$

where $\mathbf{Y}\in\mathbf{R}^{Q\times1}$ and $\mathbf{Z}\in\mathbf{R}^{Q\times M(n+1)}$ are matrices whose i-th rows are defined as y_i and $[\tilde{\mathbf{z}}_i^{1^T}\ \ \tilde{\mathbf{z}}_i^{2^T}\ \ \ldots\ \ \tilde{\mathbf{z}}_i^{M^T}]$, respectively and $\tilde{\mathbf{z}}_i^k=\psi_k(\mathbf{x}_i)\beta_k(\mathbf{x}_i)\mathbf{z}_i$.

Proof: The proof is given in Appendix A. ■

Theorem 2: *Consider a nonlinear process with a single output, $y\in\mathbf{R}$, and an n-dimensional input, $\mathbf{x}\in\mathbf{R}^{n\times1}$ and denote observed input-output data points of the process with $(\mathbf{x}_i,y_i), i=1,2,\ldots,Q$ .Consider the represented TS-DLM fuzzy model in Equation(4) for this process. It can be shown that:*

a) Suppose each $\boldsymbol{\theta}_k$ is known. $\mathbf{S}_k$, which satisfies Equation (7), is the optimum answer minimizing $J_k=\sum_{i=1}^{Q}\psi_k(\mathbf{x}_i)(y_i-\tilde{y}_i^k)^2$.

$$\sum_{i=1}^{Q}(y_i'^k)^2(\mathbf{A}_i^{kk}\otimes\mathbf{A}_i^{kk})vec(\mathbf{S}_k)=-vec(\sum_{i=1}^{Q}y_i'^k e_i'^k\mathbf{A}_i^{kk}) \tag{7}$$

where $y_i'^k=(\psi_k(\mathbf{x}_i))^{0.5}\mathbf{z}_i^T\boldsymbol{\theta}_k$ *and* $e_i'^k=(\psi_f(\mathbf{x}_i))^{0.5}y_i-y_i'^k$.

b) Suppose each $\mathbf{S}_k$ is known. The computed $\boldsymbol{\theta}_k$ (through Equation (8)) provides optimum solution to minimizing J_k.

$$\boldsymbol{\theta}_k=(\mathbf{Z}_k^T\mathbf{Z}_k)^{-1}\mathbf{Z}_k^T\mathbf{Y}_k \tag{8}$$

where $\mathbf{Y}_k\in\mathbf{R}^{Q\times1}$ and $\mathbf{Z}_k\in\mathbf{R}^{Q\times M(n+1)}$ are matrices *whose i -th rows are defined as $\breve{y}_i^k = (\psi_f(\mathbf{x}_i))^{0.5} y_i$ and $\breve{\mathbf{z}}_i^k = (\psi_k(\mathbf{x}_i))^{0.5} \beta_k(\mathbf{x}_i)\mathbf{z}_i$, respectively.*

Proof: The proof is given in Appendix B. ■

Some important notes about the above theorems and computational complexity of the answers are explained below:

1. Since $\mathbf{B}_{fk}$ and $\mathbf{A}_i^{kk}$ are symmetric matrices, $\mathbf{B}_{fk} = \mathbf{B}_{kf}$, and $\mathbf{S}_k$ is considered symmetric, approximately half of the linear equations (rows of Equations 5 and 7) are redundant. Hence, to compute $\mathbf{S}_k, k = 1,2,\ldots,M$, one must remove these redundant equations and then solve the remaining set of linear equations:
2. It is obvious from the above theorems that Equations (5) and (6) provide global optimization where Equations (7) and (8) provide local optimization. It can be easily inferred that the local optimization method offers a better computational performance and higher numerical stability in comparison with the global one. This is due to matrix inversion operating on smaller matrices. However, it is expected that global optimization provides more accurate solutions.

3. SIGMOID MFS TO LOCALIZE DLMS

Since DLMs can be optimized as nonlinear sub-models, they can supply local nonlinearities for a process. So, to achieve better optimization performance, DLMs are put in the place to learn local nonlinearities of a process where MFs are utilized mainly in fuzzy partitioning. Accordingly, we should use smooth MFs, which have the ability to represent fuzzy subspaces with rather large flat tops around the focal points.

Some MFs such as trapezoids may represent flat top regions but are not smooth. Gaussian Functions (GFs) are smooth but they vary around their focal points through exponential functions with quadratic arguments. Here, we suggest a smooth MF through sigmoid functions. These are exponential functions with first order arguments, which allow fuzzy subspaces with larger flat regions to be represented around focal points. Let $\vec{a}_j \quad j = 1,2,\ldots,n$ denote main axes of input-space. In correspondence with k -th fuzzy rule, consider k -th standard hyper-rectangle, which centered at $\mathbf{c}_k = [c_1^k \; c_2^k \ldots c_n^k]$. L_j^k denotes its length along $\vec{a}_j$. Now, with respect to k -th considered standard hyper-rectangle, k -th MF can be formulated as follows:

$$
\begin{aligned}
&\kappa_k(\mathbf{x}) = \prod\nolimits_{j=1}^n \tau_j^k(x_j) \\
&\tau_j^k(x_j) = \sigma(p_j^k)\sigma(\overline{p}_j^k) \qquad \sigma(u) = 1/(1+e^{-2\gamma u}) \\
&p_j^k = ((c_j^k + L_j^k/2) - x_j)\big/L_j^k \\
&\overline{p}_j^k = (x_j - (c_j^k - L_j^k/2))\big/L_j^k
\end{aligned}
\tag{9}
$$

where $\sigma(\cdot)$ is a sigmoid function and γ determines its variation rate. The function $\tau_j^k(\cdot)$ determines k -th fuzzy set of x_j . As it is seen from Equation (9) , k -th MF is a product of $\tau_j^k(\cdot)$s, $j = 1,2,\ldots,n$. Fig. 2(a) shows a function diagram of a $\tau_j^k(\cdot)$ along $\vec{a}_j$ with $\gamma = 8$, $c_j^k = 0.5$ and $L_j^k = 1$. As it can be seen the function is in its maximum at $c_j^k = 0.5$ (center), and reduces as it distances from its focal point, similar to a convex function, and converges to zero when x_j is far off from the center. Fig. 2(b) shows the diagram of the above defined MF with $\mathbf{c}_1^T = [0.5 \;\; 0.5]$, $L_1^1 = L_2^1 = 1$ and $\gamma = 8$. As it can be seen, it has produced a nearly flat region around the center but decreases and converges to zero when distancing from the center.

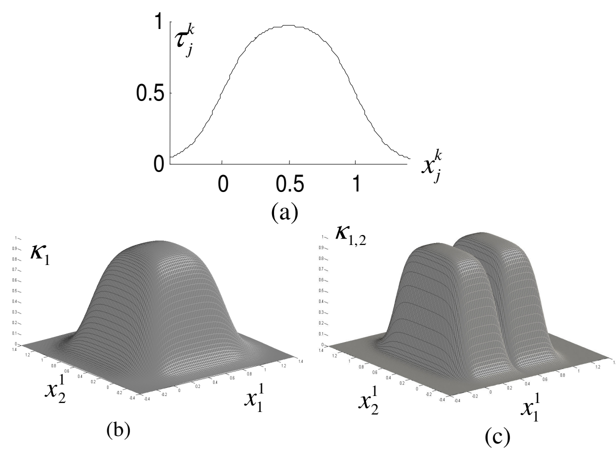

Figure 2: (a) Diagram function of a fuzzy set for x_j with $\gamma = 8$, $c_j^k = 0.5$ and $L_j^k = 1$. (b) Diagram of a 2- dimensional MF with $\mathbf{c}_1^T = [0.5 \; 0.5]$, $L_1^1 = L_2^1 = 1$ and $\gamma = 8$. (c) Diagram of two equal fuzzy subspaces, which are the results of dividing the initial one -shown at (b)- along $\vec{a}_1$.

Now, suppose we want to replace the k_w -th MF ($k_w \in [1, M]$) with two new ones: k_1 and k_2 by dividing its function diagram along axis $\vec{a}_{j^*}$ ($j^* \in [1, n]$), orthogonally. To this end, it is sufficient for the centers and lengths of two new equal hyper-rectangles k_1 and k_2 to be determined as follows (through the k_w-th considered hyper-rectangle):

$$
\begin{aligned}
& L_j^{k_1} = L_j^{k_2} = L_j^{k_w}, \; j \neq j^* \\
& L_{j^*}^{k_1} = L_{j^*}^{k_1} = L_{j^*}^{k_w} / 2 \\
& c_j^{k_1} = c_j^{k_2} = c_j^{k_w}, \; j \neq j^* \\
& c_{j^*}^{k_1} = c_{j^*}^{k_w} + L_{j^*}^{k_w} / 4, \quad c_{j^*}^{k_2} = c_{j^*}^{k_w} - L_{j^*}^{k_w} / 4
\end{aligned}
\qquad (10)
$$

If we apply Equation (9) to the new hyper-rectangles: k_1 and k_2, MFs $\kappa_{k_1}(\mathbf{x})$ and $\kappa_{k_2}(\mathbf{x})$ are achieved. Fig. 2(c) shows a plot of functions of two new MFs, which are constructed through dividing the former diagram function —shown in Fig. 2(b)— along $\vec{a}_1$.

4. An incremental identification approach for the TS-DLM fuzzy model

In this section, we present an incremental identification approach for the TS-DLM fuzzy model with introduced MFs. First, some main parts of the identification approach are explained:

1. At first a big-enough standard $n-$dimensional hyper-rectangle, which involves all observed input data points, is defined:

$$
\begin{aligned}
& \mathbf{c}_1 = \sum_{i=1}^{Q} \mathbf{x}_i / Q \\
& L_1^j = \max_i (x_i^j) - \min_i (x_i^j)
\end{aligned}
\qquad (11)
$$

where $\mathbf{c}_1$ denotes its center and L_1^j denotes its length along $\vec{a}_j$.

2. Suppose we want to define k-th DLM through its normalized MF $\psi_k(\mathbf{x}_i)$ and its known related hyper-rectangle. First, its linear parameters are computed through weighted least squares as follows:

$$
\boldsymbol{\theta}_k = (\overline{\mathbf{Z}}_k^T \overline{\mathbf{Z}}_k)^{-1} \overline{\mathbf{Z}}_k^T \mathbf{Y}_k \qquad (12)
$$

where $\mathbf{Y}_k \in \mathbf{R}^{Q\times 1}$ and $\overline{\mathbf{Z}}_k \in \mathbf{R}^{Q\times M(n+1)}$ are matrices whose $i-$th rows are defined as $\breve{y}_i^k = (\psi_k(\mathbf{x}_i))^{0.5} y_i$ and $\overline{\mathbf{z}}_i^k = (\psi_k(\mathbf{x}_i))^{0.5} \mathbf{z}_i$ respectively. Then, with respect to $\mathbf{c}_k$ and $\boldsymbol{\theta}_k$, $\mathbf{S}_k$ is computed through (7). Finally, with respect to $\mathbf{c}_k$ and $\mathbf{S}_k$, $\boldsymbol{\theta}_k$ is updated through (8).

3. In this approach, at each stage, a DLM in the TS-DLM fuzzy model is chosen and replaced with two new ones. To choose a DLM and replace it with two new ones, we need to evaluate DLMs in the TS-DLM fuzzy model. Therefore, we define local Sum of Squared Errors (SSE) index as follows.

$$
\begin{aligned}
& J_k = \sum_{i=1}^{Q} \psi_k(\mathbf{x}_i)(e_i^k)^2 \\
& e_i^k = y_i - \tilde{y}_i^k
\end{aligned}
\qquad (13)
$$

The DLM, which has the highest local SSE is chosen:

$$
k_w = \arg_k(\max_J (J_k)) \qquad (14)
$$

4. To achieve a more accurate and valid TS-DLM fuzzy model, suitable interpolations between DLMs must be done. For this purpose, we performed numerous simulations for different cases. As a result, we practically observed that choosing γ between 5 and 10 yields better results. Lower values of γ are suggested to be used in cases where more extrapolation for a process is required. In this paper, $\gamma = 8$ is chosen for the considered examples.

5. Two sets of data points are provided for train and test phases. Using the training data set, the fuzzy rules of the TS-DLM fuzzy model are learned. The algorithm stops if there is no significant reduction in SSE index. We use test data set to evaluate the identified model.

6. Through a post-tuning procedure, one can optimize the parameters of DLMs, locally, through sequential computing of Equations (7) and (8) at the end of algorithm, N times. Alternatively, one can do this globally through computing Equations (5) and (6) once.

Fig. 3 presents a flowchart of the suggested identification algorithm with details.

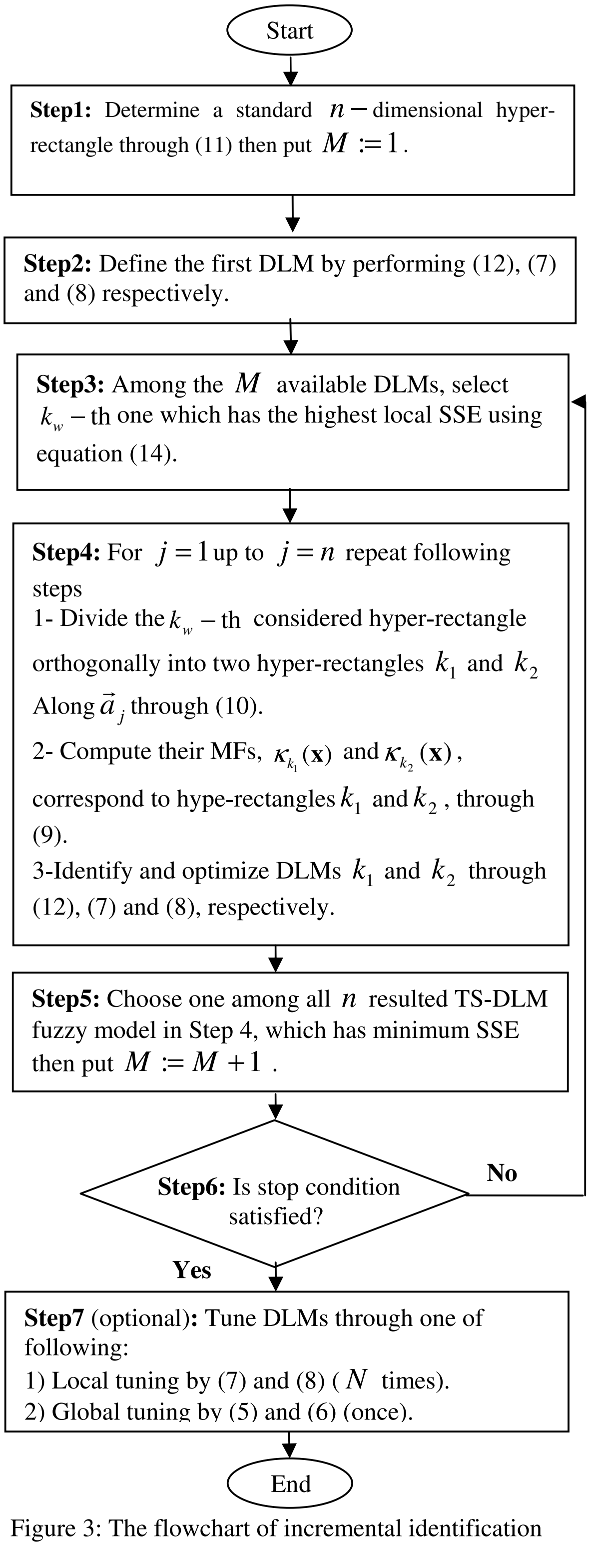

Figure 3: The flowchart of incremental identification approach proposed for the TS-DLM fuzzy model

5. Case studies

Here, we present the experimental results of applying the proposed identification algorithm to the TS-DLM fuzzy model for four different examples. The first example illustrates how DLMs are created to approximate a nonlinear function. Then, the performance of the proposed TS-DLM model is evaluated and compared in there different case studies: prediction of the MG time series, identification of a steam generator model and approximation of a nonlinear function for a sun sensor.

A. An illustrative example

Approximation of a Sinc function through the suggested fuzzy model is considered:

$$y = \mathrm{sinc}(x_1, x_2) = \frac{\sin(x_1)}{x_1}\frac{\sin(x_2)}{x_2} \qquad (15)$$

Seven hundred training data pairs and 350 test data pairs are selected randomly from the grid points of the range [-10,10]×[-10,10] within the input-space of the Sinc function. We apply the algorithm explained in Fig. 3 to identify the TS-DLM fuzzy model for the Sinc function. Fig. 4 shows (Root Mean Square Errors) RMSE plots of training and test data sets versus number of rules.

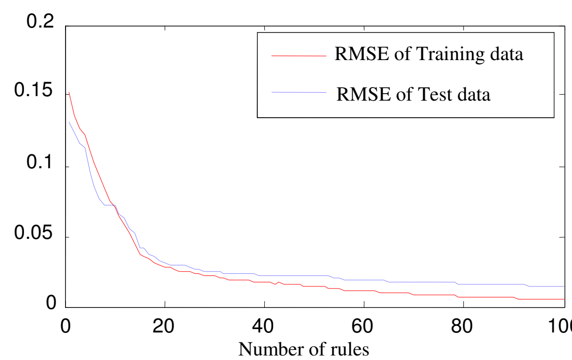

Figure 4: RMSE plots of train and test data sets versus number of rules for the Sinc function

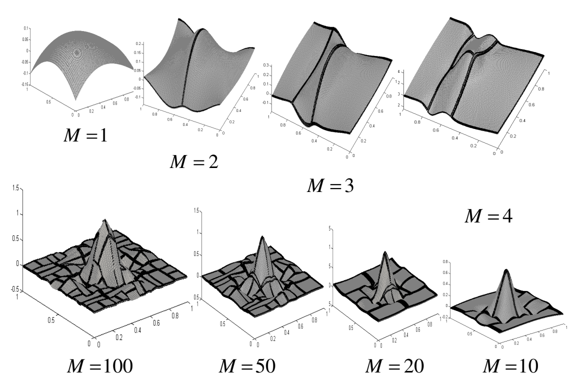

Figure 5: Some views of the approximated Sinc function with different number of DLMs.

As it is seen in Fig. 4, the RMSE index decreases when the number of rules is increased; Fig. 5 shows a few views of approximated Sinc functions with different number of DLMS and normalized input data space.

As it can be seen, the DLMs are localized in input-space through introduced MFs. They can be identified from each other by their rectangle boundaries which are marked with bold black lines. The focal points of quadratic functions ($\mathbf{c}_k$, $k=1,2,\ldots,M$) are the center points of determined rectangles. It is seen, clearly, while the number of rules in the TS-DLM fuzzy model is increased, the approximated function becomes more similar to a Sinc function.

B. Predicting chaotic dynamics (Mackey-Glass time series)

Here, the TS-DLM fuzzy model is identified to predict the Mackey-Glass (MG) time series. The MG time series is generated through a delayed differential equation defined as follows.

$$\dot{x}(t)=\frac{0.2x(t-\tau_1)}{1+x^{10}(t-\tau_1)}-0.1x(t) \qquad (16)$$

The delayed differential Equation (15) is solved by fourth-order Runge-Kutta method with time-step 0.1 from $t=0$ to $t=5500$ and the initial condition $x(0)=1.2$, as well as delay $\tau_1=17$. In this experiment, 3000 evenly sampled data points, for $t=201$ to 3200, are chosen as the training data set; 500 data points, for t=5001 to 5500, are used as the test data set. The goal is 85-step-ahead predictions of time series based on its values at the current moment along with 6th, 12th, and 18th lags.

$$\begin{aligned} y_t &= x(t+85) \\ \mathbf{x}_t &= [x(t-18), x(t-12), x(t-16), x(t)]^T \end{aligned} \qquad (17)$$

This particular selection of training and test data as well as $\mathbf{x}_t$ and y_t make our results comparable with other reported results [3]. The No Dimensional Error Index (NDEI), which is defined as the ratio of the root mean square error over the standard deviation of the target data, is considered as the criterion for evaluation and comparison purposes. The suggested TS-DLM fuzzy model with introduced MFs is identified through proposed identification algorithm for the generated MG time series.

Fig. 6 shows the RMSE plots of train and test data sets versus the number of fuzzy rules. As it is seen, the reduction rate of error index is kept on while $M<200$. However, the optimum index is met at $M=190$.

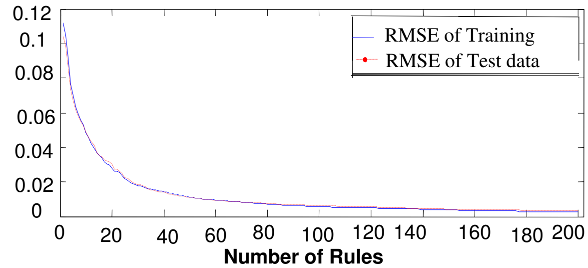

Figure 6: the RMSE plots of train and test data sets versus number of fuzzy rules of the TS-DLM used in prediction of MG time series.

Table 1 presents the MG time series prediction results, including, number of rules (neurons) , number of training epochs, training times and computed NDEI for both training and testing data. The five initial rows of given results in the table are the same with what is reported in [3]. Also, the achieved prediction results through a MATLAB implementation of NFCRMA algorithm ([9]), is presented in Table 1. For this purpose, to achieve better results, all suggested parameters are considered.

TABLE 1
PREDICTION RESULTS OF SOME NONLINEAR MODELS ON MG TRAINING AND TEST DATA; THE FIRST FIVE ROWS OF RESULTS HAD BEEN REPORTED IN [3].

Methods	Rules (Neu.)	Epochs (Iter.)	Train. NDEI	Test. NDEI
MLP-BP	60	500	0.021	0.022
ANFIS	81	200	0.028	0.029
DENFIS I	116	2	0.068	0.068
DENFIS I	883	2	0.023	0.019
DENFIS II (Mixed with MLP)	58	100	0.017	0.016
NFCRMA [9]	90	30	0.0225	0.024
TS-DLM (without tuning)	190	**-**	**0.012**	**0.015**
TS-DLM (without tuning)	70	-	0.037	0.037
TS-DLM (local tuning)	70	*N*=100	0.028	0.030
TS-DLM (global tuning)	70	-	0.019	0.026

It is shown that the suggested TS-DLM fuzzy model with $M=190$ fuzzy rules provides significant improvement over other previous achieved results. It should be noted that one of other favorable results is for DENFIS II, which its sub-models are MLP-based models. However, the identified model by DENFIS II and MLP model do not have linear interpretations like first-order TS fuzzy models. To provide a TS-DLM fuzzy model with more concise structure, we also identify it with $M=70$ fuzzy rules. As it is shown in this situation the result is favorable. However, to increase the performance of the model, local and global post-tuning is used despite their rather high training time. As it is shown in the Table 1, both post-tuning on the fuzzy model has increased the performance. Fig. 7 shows the original and predicted MG time series for considered test data points, (a), and the error signal of prediction by TS-DLM model with fuzzy rules, (b).

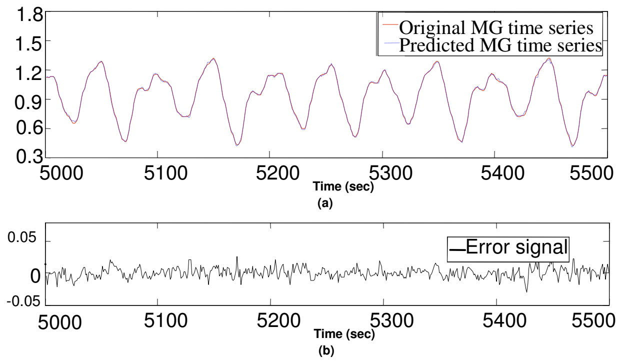

Figure 7: The original and predicted MG time series for considered test data points, (a), and the error signal of prediction by TS-DLM model with $M = 70$ fuzzy rules, (b).

To come up with a more thorough study of the performance of the TS-DLM fuzzy model, we have also identified the TS-DLM fuzzy models in various other states. Table 2 shows the prediction results for these states.

TABLE 2
PREDICTION RESULTS OF SOME OTHER STATES FOR THE TS-DLM FUZZY MODELS.

Methods	rules	Epochs	Training NDEI	Testing NDEI
TS-DLM (without tuning)	140	-	0.018	0.020
TS-DLM (local tuning)	140	N=100	0.014	0.015
TS-DLM (global tuning)	140	-	0.0065	0.1652
TS-DLM with Gaussian MFs	70	-	0.055	0.062
TS-DLM with linear submodels	70	-	0.093	0.10

Although performing global post-tuning has increased the performance for the model with $M = 70$ fuzzy rules, this may be unsuitable when the number of fuzzy rules is rather high. To illustrate the situation, we identify the TS-DLM fuzzy model with $M = 140$ fuzzy rules. As it is shown in Table 2, comparing the favorable results obtained for local-tuning, global post-tuning shows unsuitable for the model with $M = 140$ fuzzy rules due to over-fitting problem. Also, to evaluate the sigmoid MF, we have identified the TS-DLM fuzzy model with a conventional MF, i.e., standard Gaussian function, hence, instead of using equation (9) in the identification algorithm, the following function is considered:

$$\kappa_k(\mathbf{x}) = \prod_{j=1}^{n} \tau_j^k(x_j)$$
$$\tau_j^k(x_j) = \exp(-(\alpha(x_j - c_j^k)/L_j^k)^2) \tag{18}$$

where $\alpha = 3$ provides a favorable interpolation coefficient [16]. It can be inferred from Table 2 that sigmoid MFs provide better performance for the TS-DLM model. Also, it can be seen from Table 2, if only linear parts of DLMs are considered, or equally, $\mathbf{S}_k = \mathbf{0},\ k = 1,2,\ldots,M$, the performance of the model decreases considerably. Two last results demonstrate the high impact of sigmoid MFs and quadratic functions in improving the performance of the suggested fuzzy model.

C. Identifying a model for a steam generator

In this case study, we identify a steam generator (dynamic) model through the introduced learning algorithm of the TS-DLM fuzzy model. Then, the performance of the TS-DLM fuzzy model is compared with those of the fuzzy models identified through MATLAB implementations of MLP-BP, ANFIS, DENFIS I, DENFIS II and NFCRMA methods. To achieve better results, all suggested parameter adjustments are considered. The data comes from DaISy: Database for the Identification of Systems, [24]. The considered output of model, y_t , is steam flow (Kg./s). This model has seven inputs: fuel, air, reference level, disturbance and three lags of steam flow: $(y_{t-1}, y_{t-2}, y_{t-3})$. Totally, 9600 input-output data point are sampled with sampling rate 3 sec. We assign 75% of these data points (initially-observed part) for training data and the rest 25% is assigned to test data. In this case, for all considered methods, we end the training phase of the models before occurrence of the over-fitting problem. The training and test RMSE plots versus number of rules are shown in Fig. 8. As it is seen, at M=8 fuzzy rules — before the over-fitting problem intensifies— the training phase for TS-DLM is terminated.

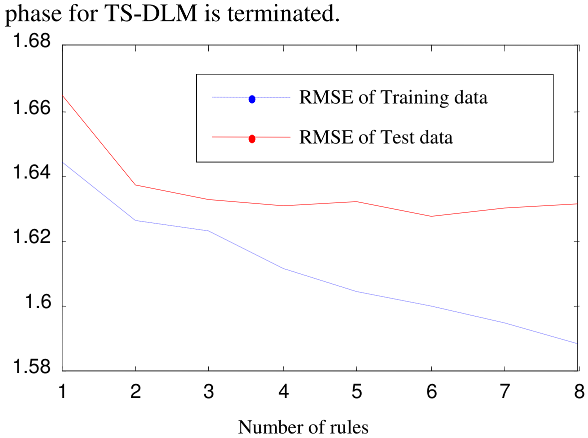

Figure 8: The RMSE plots of train and test data sets versus number of fuzzy rules of the TS-DLM fuzzy model identified

for the steam generator model.

Table 3 presents the results including number of rules, epochs, training time and the achieved NDEI for training and test data.

TABLE 3
THE IDENTIFICATION RESULTS OF SOME NONLINEAR IDENTIFICATION APPROACHES FOR THE STREAM GENERATOR MODEL

Methods	Rules (Neu.)	Epoch. (Iter.)	Train. NDEI	Test. NDEI	Train. Time (sec)
MLP-BP	20	200	0.021	0.0240	292
ANFIS	29	10	0.0186	0.0213	232
DENFIS I	34	10	0.0190	0.0210	364
DENFIS II	34	10	0.0188	0.0240	861
NFCRMA [9]	15	30	0.0189	0.0209	695
TS-DLM (without tuning)	8	-	0.0186	0.0204	151
TS-DLM (local tuning)	8	N=10	0.0185	0.0204	195
TS-DLM (global tuning)	8	1	0.0185	0.0206	216

As it is seen, the resulting TS-DLM model, with its much lower number of fuzzy rules, has favorable generalization capability in comparison with fuzzy models resulting through other approaches. However, the global tuning makes the testing NDEI worse due to over-fitting problem.

D. Approximation of a nonlinear function for a sun sensor

Due to memory limitations in embedded systems, it is demanded that the large look-up tables of sensors or actuators be replaced by accurate-enough nonlinear models with concise structures. Here, we apply the proposed TS-DLM with its learning algorithm in Fig. 3 to approximate the look-up table of a sun sensor utilized in attitude control of satellites (courtesy of Space Research Center of Iran for the use of data). This look-up table has 110 columns and 110 rows. Considering the indices of the rows and columns as inputs, a 3-dimensional surface for the look-up table is formed and is shown in Fig. 9.

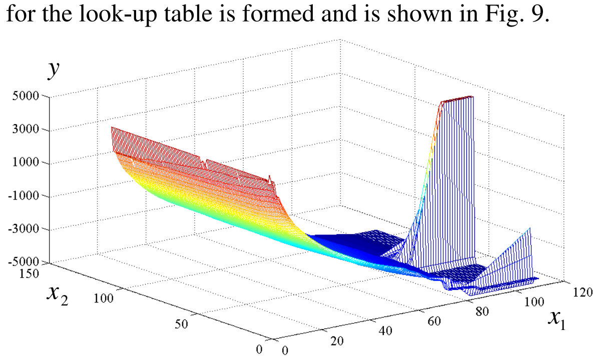

Figure 9: A 3- dimensional surface of the look- up table of a sun sensor (the variables x_1 and x_2 are indices of rows and columns of the table and y is t).

Due to the rough region appearing in the last 30 rows of the table, they are memorized directly and the rest of the look-up table is considered for approximation. The maximum absolute difference of the estimated and original values of the table, Max, is considered for evaluation. In this case, the accuracy of the model providing $Max < 900$ is satisfactory. We apply all mentioned methods in Table 3 to identify a model for the current case study. In this case, there are no evaluation phase and only a model is trained to approximate the look-up table. Table 4 shows the achieved modeling and approximation results of different methods.

TABLE 4
THE MODELING AND APPROXIMATION RESULTS FOR THE LOOK-UP TABLE OF THE SUN SENSOR.

Methods	Rules (Neu.)	Epoch. (Iter.)	Train. NDEI	Max	Train. Time (sec)
MLP-BP	15	100	0.0562	801	168
ANFIS	10	50	0.0368	706	61
DENFIS I	32	15	0.081	871	311
DENFIS II	32	50	0.063	802	721
NFCRMA [9]	20	30	0.0346	726	614
TS-DLM (without tuning)	6	-	0.0445	762	20
TS-DLM (local tuning)	6	N=20	0.031	753	71
TS-DLM (global tuning)	6	-	0.037	470	46

TABLE 5.
PREMISE AND CONSEQUENCE PARAMETERS OF THE RULES OF THE IDENTIFIED TS-DLM FUZZY MODEL.

Rule	$\mathbf{c}_k$	$\mathbf{L}_k$	$\mathbf{S}_k$	$\boldsymbol{\theta}_k$
1	$\begin{bmatrix} 0.75 \\ 0.75 \end{bmatrix}$	$\begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$	$\begin{bmatrix} 0.0181 & -0.042 \\ -0.042 & 0.0781 \end{bmatrix}$	$\begin{bmatrix} 455 \\ -3865 \\ -13 \end{bmatrix}$
2	$\begin{bmatrix} 0.5 \\ 0.375 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0.25 \end{bmatrix}$	$\begin{bmatrix} 0.3275 & 0.0823 \\ 0.0823 & -0.002 \end{bmatrix}$	$\begin{bmatrix} -324 \\ -6303 \\ 1580 \end{bmatrix}$
3	$\begin{bmatrix} 0.5 \\ 0.1875 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0.125 \end{bmatrix}$	$\begin{bmatrix} 0.5615 & 0.3194 \\ 0.3194 & 10.704 \end{bmatrix}$	$\begin{bmatrix} -300 \\ -9498 \\ 2435 \end{bmatrix}$
4	$\begin{bmatrix} 0.5 \\ 0.9375 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0.625 \end{bmatrix}$	$\begin{bmatrix} -0.122 & -0.629 \\ -0.629 & -18.691 \end{bmatrix}$	$\begin{bmatrix} -52 \\ -1565 \\ 341 \end{bmatrix}$
5	$\begin{bmatrix} 0.25 \\ 0.75 \end{bmatrix}$	$\begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$	$\begin{bmatrix} 0.579 & 0.1405 \\ 0.1405 & -0.0643 \end{bmatrix}$	$\begin{bmatrix} -743 \\ -4207 \\ 543 \end{bmatrix}$
6	$\begin{bmatrix} 0.5 \\ 0.3125 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0.625 \end{bmatrix}$	$\begin{bmatrix} 0.1197 & 1.3610 \\ 1.361 & -49.01 \end{bmatrix}$	$\begin{bmatrix} -83 \\ -3191 \\ 453 \end{bmatrix}$

As it is seen in Table 4, the identified TS-DLM fuzzy model with global-tuning has the least number of fuzzy rules and best Max index, and the identified TS-DLM fuzzy model with local-tuning has the best NDEI index. Table 5 presents premise and consequence parameters of the rules of the identified TS-DLM fuzzy model. Note that the considered input and output data points of the model have been normalized.

6. Conclusion

In this paper, we proposed a new high order TS fuzzy model as an alternative for identification of complicated processes. Accordingly, Deformed Linear Models (DLMs) were suggested to be used as consequence parts of fuzzy rules. A DLM was defined as the product of a linear model and a quadratic function. It was explained that the resulting high-order TS-DLM fuzzy model can be interpreted like the first-order one, which facilitates its development in many areas like control applications. Also, through Theorem 1 and Theorem 2, we proved that linear or nonlinear models in DLMs can be optimized independently in two local and global manners. A fitting membership function for suggested fuzzy model, which is the product of some sigmoid functions, was introduced. Then, an incremental identification algorithm was proposed for the TS-DLM fuzzy model. At each stage, the DLM with highest local SSE was replaced with two new optimized ones. Simultaneously, one fuzzy rule was replaced with two new ones. Through an illustrative example, it was shown how DLMs in the TS-DLM fuzzy model were shaped to approximate a Sinc function. The proposed model was applied to predict the Mackey-Glass time series, to identify a steam generator model and to approximate a nonlinear function for a sun sensor. The comparison of the results with some other approaches demonstrated high accuracy and validity of the proposed model in considered prediction, identification and approximation case studies. It was further shown that one can provide a tradeoff among a smallness of the structure, accuracy, training times and over-fitting for this model by determining the number of fuzzy rules and performing post-tuning methods.

7. Acknowledgment

The authors would like to thank Mohammad M. Ajallooeian for reading an initial draft of this paper and providing many helpful comments.

8. References

[1] T. Takagi, and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Trans. Syst., Man Cybern.*, vol. 15, pp. 116-132, 1985.

[2] R. Jang, "ANFIS: Adaptive network-based fuzzy inference system," *IEEE Trans. Syst., Man, Cybern.*, vol. 23, pp. 665-685, 1993.

[3] N. Kasabov, "DENFIS: Dynamic Evolving Neural-Fuzzy Inference System and its application for time-series prediction," *IEEE Trans. Fuzzy Syst.*, vol. 10, pp. 144-154, 2002.

[4] Y. Chen, B. Yang, A. Abraham, and L. Peng, "Automatic design of hierarchical Takagi–Sugeno type fuzzy systems using evolutionary algorithms," *IEEE Trans. Fuzzy Syst*., vol. 15, pp. 385-397, 2007.

[5] J. Abonyi, R. Babuska, and F. Szeifert, "Modified Gath-Geva fuzzy clustering for identification of Takagi-Sugeno fuzzy models," *IEEE Trans. Syst. Man and Cybern.—PART B*, vol. 32, pp. 612-621, 2002.

[6] P. Pulkkinen, and H. Koivisto, "Identification of interpretable and accurate fuzzy classifiersand function estimators with hybrid methods," *Applied Soft Computing*, vol. 7, pp. 520-533, 2007.

[7] C. F. Juang, "A self-organizing TS-type fuzzy network with support vector learning and its application to classification problems," *IEEE Trans. Fuzzy Syst*., vol. 15, pp. 998-1008, 2007.

[8] A. Kalhor, B. N. Araabi, and C. Lucas, "Online identification of a neuro-fuzzy model through indirect fuzzy clustering of data space," FUZZ-IEEE2009, the 18th Int. conference on fuzzy systems, 21-24 Aug., Korea, 2009, pp. 356-359.

[9] C. Li, J. Zhou, X. Xiang, Q. Li and X. An, "T–S fuzzy model identification based on a novel fuzzy c-regression model clustering algorithm," *Engineering Applications of Artificial Intelligence*, vol. 22, pp. 646–653, 2009.

[10] C .F. Juang, "A TSK-Type recurrent fuzzy network for dynamic systems processing by neural network and genetic algorithms," *IEEE Trans. Fuzzy Syst*., vol. 10, pp. 155-170, 2008.

[11] H. Du, and N. Zhang, "Application of evolving Takagi–Sugeno fuzzy model to nonlinear system identification," *Applied soft computing*, vol. 8, pp. 876-686, 2007.

[12] L. Zhaoa , F. Qiana, Y. Yangb, Y. Zengb, and H. Sub, "Automatically extracting T–S fuzzy models using cooperative random learning particle swarm optimization," *Applied Soft Computing* , vol. 10, pp. 938-944, 2010.

[13] P.A. Mastorocostas, and J.B. Theocharis, "A recurrent fuzzy-neural model for dynamic system identification," *IEEE Trans. Syst. Man Cybern.,* vol. 32, pp. 176-190, 2002.

[14] C. Li, and K. H. Cheng, " Recurrent neuro-fuzzy hybrid-learning approach to accurate system modelin," *Fuzzy Sets and Systems,* vol. 58, pp. 194-122, 2007.

[15] A. Savran, "An adaptive recurrent fuzzy system for nonlinear identification," *Applied Soft Computing*,vol. 7, pp. 593-600, 2007.

[16] O. Nelles, *Nonlinear System Identification*, New York: Springer, 2001, p. 365.

[17] A. Kalhor, B.N. Araabi, and C. Lucas, "A New Split and Merge Algorithm for Structure Identification in Takagi-Sugeno Fuzzy Model," in *Proc. of 7th Int. Conf. on Intelligent Systems Design and Applications*, 2007, pp. 258-261.

[18] A. Kalhor, B.N. Araabi, and C. Lucas, "An Online Predictor Model as Adaptive Habitually Linear and Transiently Nonlinear Model," *Evolving Systems*, vol. 1, pp. 29-41, 2010.

[19] B. Rezaee, and M.H.F. Zarandi, "Data-driven fuzzy modeling for Takagi–Sugeno–Kang fuzzy system," *Information Sciences*, vol. 180, pp. 241-255, 2010.

[20] J.B. Theocharis, "A high-order recurrent neuro-fuzzy system with internal dynamics: Application to the adaptive noise cancellation," *Fuzzy Sets and Systems*, vol. 157, pp. 471-500, 2006.

[21] Z.L. Sun, K. F. Au, and T. M. Choi, "A neuro-fuzzy inference system through integration of fuzzy logic and extreme learning machines," *IEEE Trans. Syst. Man Cybern.—PART B*, vol. 37, pp. 1321-1331, 2007.

[22] J. N. Choi, S. K. Oh and W. Pedrycz, "Identification of fuzzy models using a successive tuning method with a variant identification ratio," *Fuzzy Sets and Systems*, vol. 159, pp. 2873-2889, 2008.

[23] K. B. Petersen, and M. S. Pedersen, "The matrix cookbook," Available: http://matrixcookbook.com, Version: Nov. 14, 2008.

[24] De Moor B.L.R. (ed.) , "DaISy: Database for the Identification of Systems," *Department of Electrical Engineering,* ESAT/SISTA, K.U.Leuven, Belgium, Available: http://www.esat.kuleuven.ac.be/sista/daisy.

9. APPENDIXES

A. *Proof of Theorem 1*

a)

$$\begin{aligned} J &= \sum_{i=1}^{Q} (y_i - \hat{y}_i)^2 \\ &= \sum_{i=1}^{Q} (y_i - \sum_{k=1}^{M} \psi_k(\mathbf{x}_i)\mathbf{z}_i^T \boldsymbol{\theta}_k (1-(\mathbf{x}_i-\mathbf{c}_k)^T \mathbf{S}_k (\mathbf{x}_i - \mathbf{c}_k))^2 \end{aligned} \quad \text{(A1)}$$

Let $\bar{y}_i^k = \psi_k(\mathbf{x}_i)\mathbf{z}_i^T\boldsymbol{\theta}_k$ and $\tilde{\mathbf{x}}_i^k = \mathbf{x}_i - \mathbf{c}_k$ then it can be shown that

$$\begin{aligned} J &= \sum_{i=1}^{Q} (y_i - \sum_{k=1}^{M} \bar{y}_i^k (1 - \tilde{\mathbf{x}}_i^{k^T} \mathbf{S}_k \tilde{\mathbf{x}}_i^k))^2 \\ &= \sum_{i=1}^{Q} (y_i^2 - 2y_i \sum_{k=1}^{M} \bar{y}_i^k (1 - \tilde{\mathbf{x}}_i^{k^T} \mathbf{S}_k \tilde{\mathbf{x}}_i^k) + \\ &\quad + (\sum_{k=1}^{M} \bar{y}_i^k (1 - \tilde{\mathbf{x}}_i^{k^T} \mathbf{S}_k \tilde{\mathbf{x}}_i^k))(\sum_{j=1}^{M} \bar{y}_i^j (1 - \tilde{\mathbf{x}}_i^{j^T} \mathbf{S}_j \tilde{\mathbf{x}}_i^j)) \\ &= \underbrace{\sum_{i=1}^{Q} (y_i^2 - 2y_i \sum_{k=1}^{M} \bar{y}_i^k + \sum_{k=1}^{M} \bar{y}_i^k \sum_{j=1}^{M} \bar{y}_i^j)}_{J_0} \\ &\quad + \sum_{i=1}^{Q} \Big(2y_i \sum_{k=1}^{M} \bar{y}_i^k \tilde{\mathbf{x}}_i^{k^T} \mathbf{S}_k \tilde{\mathbf{x}}_i^k \\ &\qquad - \sum_{k=1}^{M} \sum_{j=1}^{M} \bar{y}_i^k \bar{y}_i^j (\tilde{\mathbf{x}}_i^{k^T} \mathbf{S}_k \tilde{\mathbf{x}}_i^k + \tilde{\mathbf{x}}_i^{j^T} \mathbf{S}_j \tilde{\mathbf{x}}_i^j) \\ &\qquad + \sum_{k=1}^{M} \sum_{j=1}^{M} \bar{y}_i^k \bar{y}_i^j (\tilde{\mathbf{x}}_i^{k^T} \mathbf{S}_k \tilde{\mathbf{x}}_i^k \tilde{\mathbf{x}}_i^{j^T} \mathbf{S}_j \tilde{\mathbf{x}}_i^j) \Big) \\ &= J_0 + \sum_{i=1}^{Q} \Big(2y_i \sum_{k=1}^{M} \bar{y}_i^k \tilde{\mathbf{x}}_i^{k^T} \mathbf{S}_k \tilde{\mathbf{x}}_i^k \\ &\qquad - \sum_{k=1}^{M} \sum_{j=1}^{M} \bar{y}_i^k \bar{y}_i^j (\tilde{\mathbf{x}}_i^{k^T} \mathbf{S}_k \tilde{\mathbf{x}}_i^k + \tilde{\mathbf{x}}_i^{j^T} \mathbf{S}_j \tilde{\mathbf{x}}_i^j) \\ &\qquad + \sum_{k=1}^{M} \sum_{j=1}^{M} \bar{y}_i^k \bar{y}_i^j (\tilde{\mathbf{x}}_i^{k^T} \mathbf{S}_k \tilde{\mathbf{x}}_i^k \tilde{\mathbf{x}}_i^{j^T} \mathbf{S}_j \tilde{\mathbf{x}}_i^j) \Big) \end{aligned} \quad \text{(A2)}$$

We use the tracing operation as follows:

$$\begin{aligned} J &= J_0 + \sum_{i=1}^{Q} \Big(2y_i \sum_{k=1}^{M} \bar{y}_i^k trace\,(\tilde{\mathbf{x}}_i^{k^T} \mathbf{S}_k \tilde{\mathbf{x}}_i^k) \\ &\qquad - \sum_{k=1}^{M} \sum_{j=1}^{M} \bar{y}_i^k \bar{y}_i^j trace\,(\tilde{\mathbf{x}}_i^{k^T} \mathbf{S}_k \tilde{\mathbf{x}}_i^k + \tilde{\mathbf{x}}_i^{j^T} \mathbf{S}_j \tilde{\mathbf{x}}_i^j) \\ &\qquad + \sum_{k=1}^{M} \sum_{j=1}^{M} trace\,(\bar{y}_i^k \bar{y}_i^j (\tilde{\mathbf{x}}_i^{k^T} \mathbf{S}_k \tilde{\mathbf{x}}_i^k \tilde{\mathbf{x}}_i^{j^T} \mathbf{S}_j \tilde{\mathbf{x}}_i^j)) \Big) \\ &= J_0 + \sum_{i=1}^{Q} \Big(2y_i \sum_{k=1}^{M} \bar{y}_i^k trace\,(\tilde{\mathbf{x}}_i^k \tilde{\mathbf{x}}_i^{k^T} \mathbf{S}_k) \\ &\qquad - \sum_{k=1}^{M} \sum_{j=1}^{M} \bar{y}_i^k \bar{y}_i^j trace\,(\tilde{\mathbf{x}}_i^{k^T} \tilde{\mathbf{x}}_i^k \mathbf{S}_k + \tilde{\mathbf{x}}_i^{j^T} \tilde{\mathbf{x}}_i^j \mathbf{S}_j) \\ &\qquad + \sum_{k=1}^{M} \sum_{j=1}^{M} trace\,(\bar{y}_i^k \bar{y}_i^j (\tilde{\mathbf{x}}_i^j \tilde{\mathbf{x}}_i^{k^T} \mathbf{S}_k \tilde{\mathbf{x}}_i^k \tilde{\mathbf{x}}_i^{j^T} \mathbf{S}_j)) \Big) \end{aligned}$$

$$\text{(A3)}$$

Let $\mathbf{A}_i^{kj} = \tilde{\mathbf{x}}_i^k \tilde{\mathbf{x}}_i^{j^T}$ then (A.3) can be rewritten as follow :

$$\begin{aligned} J = J_0 &+ \sum_{i=1}^{Q} \Big(2y_i \sum_{k=1}^{M} \bar{y}_i^k trace\,(\mathbf{A}_i^{kk} \mathbf{S}_k) \\ &\qquad - \sum_{k=1}^{M} \sum_{j=1}^{M} \bar{y}_i^k \bar{y}_i^j trace\,((\mathbf{A}_i^{kk} \mathbf{S}_k + \mathbf{A}_i^{jj} \mathbf{S}_j)) \\ &\qquad + \sum_{k=1}^{M} \sum_{j=1}^{M} \bar{y}_i^k \bar{y}_i^j trace\,(((\mathbf{A}_i^{jk} \mathbf{S}_k \mathbf{A}_i^{kj} \mathbf{S}_j)) \Big) \end{aligned} \quad \text{(A4)}$$

Here, we compute the extremums of J :

$$\begin{aligned} &\frac{\partial J}{\partial \mathbf{S}_f} = 0 \\ &\Rightarrow \sum_{i=1}^{Q} \Big(2y_i \bar{y}_i^f \mathbf{A}_i^{ff} - 2\sum_{k=1}^{M} \bar{y}_i^k \bar{y}_i^f \mathbf{A}_i^{ff} + 2\sum_{k=1}^{M} \bar{y}_i^k \bar{y}_i^f \mathbf{A}_i^{kf} \mathbf{S}_k \mathbf{A}_i^{fk} \Big) = 0 \\ &\Rightarrow 2\sum_{i=1}^{Q} \sum_{k=1}^{M} \bar{y}_i^k \bar{y}_i^f (\mathbf{A}_i^{kf} \mathbf{S}_k \mathbf{A}_i^{fk}) = 2\sum_{i=1}^{Q} \Big(\sum_{k=1}^{M} \bar{y}_i^k \bar{y}_i^f \mathbf{A}_i^{ff} - y_i \bar{y}_i^f \mathbf{A}_i^{ff} \\ &\Rightarrow \sum_{k=1}^{M} \sum_{i=1}^{Q} \bar{y}_i^k \bar{y}_i^f (\mathbf{A}_i^{kf} \mathbf{S}_k \mathbf{A}_i^{fk}) = \sum_{i=1}^{Q} \mathbf{A}_i^{ff} \bar{y}_i^f (\underbrace{\sum_{k=1}^{M} \bar{y}_i^k}_{\bar{y}_i} - y_i) \qquad \text{(A5)} \\ &\Rightarrow \sum_{k=1}^{M} \sum_{i=1}^{Q} \bar{y}_i^k \bar{y}_i^f (\mathbf{A}_i^{kf} \mathbf{S}_k \mathbf{A}_i^{fk}) = \sum_{i=1}^{Q} \mathbf{A}_i^{ff} \bar{y}_i^f (\bar{y}_i - y_i) \\ &\Rightarrow \sum_{k=1}^{M} \sum_{i=1}^{Q} \bar{y}_i^k \bar{y}_i^f (\mathbf{A}_i^{kf} \mathbf{S}_k \mathbf{A}_i^{fk}) = -\sum_{i=1}^{Q} \mathbf{A}_i^{ff} \bar{y}_i^f (y_i - \bar{y}_i) \\ &\Rightarrow \sum_{k=1}^{M} \sum_{i=1}^{Q} \bar{y}_i^k \bar{y}_i^f (\mathbf{A}_i^{kf} \mathbf{S}_k \mathbf{A}_i^{fk}) = -\sum_{i=1}^{Q} \mathbf{A}_i^{ff} \bar{y}_i^f \bar{e}_i \end{aligned}$$

We can rewrite the above equation as follows:

$$\sum_{k=1}^{M} \underbrace{\sum_{i=1}^{Q} \bar{y}_i^k \bar{y}_i^f (\mathbf{A}_i^{kf} \otimes \mathbf{A}_i^{kf})}_{\mathbf{B}_{fk}} vec(\mathbf{S}_k) = \underbrace{-vec(\sum_{i=1}^{Q} \mathbf{A}_i^{ff} \bar{y}_i^f \bar{e}_i)}_{\mathbf{C}_f}$$

$$\Rightarrow \begin{bmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} & \cdots & \mathbf{B}_{1M} \\ \mathbf{B}_{21} & \mathbf{B}_{22} & \cdots & \mathbf{B}_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{B}_{M1} & \mathbf{B}_{M2} & \ldots & \mathbf{B}_{MM} \end{bmatrix} \begin{bmatrix} vec(\mathbf{S}_1) \\ vec(\mathbf{S}_2) \\ \vdots \\ vec(\mathbf{S}_M) \end{bmatrix} = \begin{bmatrix} \mathbf{C}_1 \\ \mathbf{C}_2 \\ \vdots \\ \mathbf{C}_M \end{bmatrix} \quad \text{(A6)}$$

Also since $\partial^2 J / \partial \mathbf{S}_f^2 = 2\sum_{i=1}^{Q} (\bar{y}_i^k)^2 \mathbf{A}_i^{ff} \mathbf{A}_i^{ff}$ is a positive definite matrix, $\mathbf{S}_k$, $k = 1,2,\ldots,M$, which satisfy (A6), minimize J .

b)

Let $\tilde{\mathbf{z}}_i^k = \psi_k(\mathbf{x}_i)(1-(\mathbf{x}_i - \mathbf{c}_k)^T \mathbf{S}_k (\mathbf{x}_i - \mathbf{c}_k))\mathbf{z}_i$; regarding to (A.1) we have

$$\begin{aligned} J &= \sum_{i=1}^{Q} (y_i - \sum_{k=1}^{M} \tilde{\mathbf{z}}_i^{k^T} \boldsymbol{\theta}_k)^2 \\ &= \sum_{i=1}^{Q} (y_i - \sum_{k=1}^{M} \boldsymbol{\theta}_k^T \tilde{\mathbf{z}}_i^k)^2 \\ &= \sum_{i=1}^{Q} (y_i - \underbrace{\begin{bmatrix} \boldsymbol{\theta}_1^T & \boldsymbol{\theta}_2^T & \ldots & \boldsymbol{\theta}_M^T \end{bmatrix}}_{\zeta^T} \underbrace{\begin{bmatrix} \tilde{\mathbf{z}}_i^{1^T} & \tilde{\mathbf{z}}_i^{2^T} & \ldots & \tilde{\mathbf{z}}_i^{M^T} \end{bmatrix}^T}_{\boldsymbol{\delta}})^2 \qquad \text{(A7)} \\ &= \sum_{i=1}^{Q} (y_i - \zeta^T \boldsymbol{\delta}_i)^2 \end{aligned}$$

As it can be known from (A.7), the problem of computing optimum ζ is the same least square optimization problem. Therefore, let $\mathbf{Z}$ denote a data matrix whose i -th row is $\boldsymbol{\delta}_i$ and $\mathbf{Y}$ denote an output vector whose i -th element is y_i . We know from least squares optimization technique that the optimum answer for ζ is computed as follows:

$$\zeta = (\mathbf{Z}^T \mathbf{Z})^{-1} \mathbf{Z}^T \mathbf{Y} \quad \text{(A8)}$$

We can get $\boldsymbol{\theta}_k$ through resulting ζ from (A.8), easily.

■

B. *Proof of Theorem 2*

a)

$$
\begin{aligned}
J_k &= \sum_{i=1}^{Q} \psi_k(\mathbf{x}_i)(y_i - \tilde{y}_i)^2 \\
&= \sum_{i=1}^{Q} \psi_k(\mathbf{x}_i)(y_i - \mathbf{z}_i^T \boldsymbol{\theta}_k (1 - (\mathbf{x}_i - \mathbf{c}_k)^T \mathbf{S}_k (\mathbf{x}_i - \mathbf{c}_k)))^2 \\
&= \sum_{i=1}^{Q} ((\psi_k(\mathbf{x}_i))^{0.5} y_i - (\psi_k(\mathbf{x}_i))^{0.5} \mathbf{z}_i^T \boldsymbol{\theta}_k (1 - (\mathbf{x}_i - \mathbf{c}_k)^T \mathbf{S}_k (\mathbf{x}_i - \mathbf{c}_k)))^2
\end{aligned}
\quad \text{(B1)}
$$

Let $\breve{y}_i^k = (\psi_k(\mathbf{x}_i))^{0.5} y_i$ and $y_i'^k = (\psi_k(\mathbf{x}_i))^{0.5} \mathbf{z}_i^T \boldsymbol{\theta}_k$

$$
\begin{aligned}
J_k &= \sum_{i=1}^{Q} (\breve{y}_i^k - y_i'^k (1 - (\mathbf{x}_i - \mathbf{c}_k)^T \mathbf{S}_k (\mathbf{x}_i - \mathbf{c}_k)))^2 \\
&= \underbrace{\sum_{i=1}^{Q} (\breve{y}_i^k - y_i'^k)^2}_{J_k^0} + \sum_{i=1}^{Q} 2\underbrace{(\breve{y}_i^k - y_i'^k)}_{e_i'^k} y_i'^k (\mathbf{x}_i - \mathbf{c}_k)^T \mathbf{S}_k (\mathbf{x}_i - \mathbf{c}_k) \\
&\qquad + \sum_{i=1}^{Q} (y_i'^k)^2 ((\mathbf{x}_i - \mathbf{c}_k)^T \mathbf{S}_k (\mathbf{x}_i - \mathbf{c}_k))^2 \\
&= J_k^0 + \sum_{i=1}^{Q} 2 y_i'^k e_i'^k (\mathbf{x}_i - \mathbf{c}_k)^T \mathbf{S}_k (\mathbf{x}_i - \mathbf{c}_k) \\
&\qquad + \sum_{i=1}^{Q} (y_i'^k)^2 ((\mathbf{x}_i - \mathbf{c}_k)^T \mathbf{S}_k (\mathbf{x}_i - \mathbf{c}_k))^2 \\
&= J_k^0 + \sum_{i=1}^{Q} 2 y_i'^k e_i'^k \, trace((\mathbf{x}_i - \mathbf{c}_k)^T \mathbf{S}_k (\mathbf{x}_i - \mathbf{c}_k)) \\
&\qquad + \sum_{i=1}^{Q} (y_i'^k)^2 \, trace((\mathbf{x}_i - \mathbf{c}_k)^T \mathbf{S}_k (\mathbf{x}_i - \mathbf{c}_k)(\mathbf{x}_i - \mathbf{c}_k)^T \mathbf{S}_k (\mathbf{x}_i - \mathbf{c}_k)) \\
&= J_k^0 + \sum_{i=1}^{Q} 2 y_i'^k e_i'^k \, trace((\mathbf{x}_i - \mathbf{c}_k)(\mathbf{x}_i - \mathbf{c}_k)^T \mathbf{S}_k) \\
&\qquad + \sum_{i=1}^{Q} (y_i'^k)^2 \, trace((\mathbf{x}_i - \mathbf{c}_k)(\mathbf{x}_i - \mathbf{c}_k)^T \mathbf{S}_k (\mathbf{x}_i - \mathbf{c}_k)(\mathbf{x}_i - \mathbf{c}_k)^T \mathbf{S}_k) \\
&= J_k^0 + \sum_{i=1}^{Q} 2 y_i'^k e_i'^k \, trace(\mathbf{A}_i^{kk} \mathbf{S}_k) + \sum_{i=1}^{Q} (y_i'^k)^2 \, trace(\mathbf{A}_i^{kk} \mathbf{S}_k \mathbf{A}_i^{kk} \mathbf{S}_k)
\end{aligned}
\quad \text{(B2)}
$$

Here, we compute the extremum of J as follows:

$$
\begin{aligned}
&\frac{\partial J_k}{\partial \mathbf{S}_k} = 0 \\
&\Rightarrow \sum_{i=1}^{Q} 2 y_i'^k e_i'^k \mathbf{A}_i^{kk} + \sum_{i=1}^{Q} 2 (y_i'^k)^2 \mathbf{A}_i^{kk} \mathbf{S}_k \mathbf{A}_i^{kk} = 0 \\
&\Rightarrow vec(\sum_{i=1}^{Q} y_i'^k e_i'^k \mathbf{A}_i^{kk}) + \sum_{i=1}^{Q} (y_i'^k)^2 (\mathbf{A}_i^{kk} \otimes \mathbf{A}_i^{kk}) vec(\mathbf{S}_k) = 0 \\
&\Rightarrow \sum_{i=1}^{Q} (y_i'^k)^2 (\mathbf{A}_i^{kk} \otimes \mathbf{A}_i^{kk}) vec(\mathbf{S}_k) = -vec(\sum_{i=1}^{Q} y_i'^k e_i'^k \mathbf{A}_i^{kk})
\end{aligned}
\quad \text{(B3)}
$$

Since $\partial^2 J_k / \partial \mathbf{S}^2 = 2\sum_{i=1}^{Q} (y_i'^k)^2 \mathbf{A}_i^{ff} \mathbf{A}_i^{ff}$ is a positive definitive matrix, the achieved $\mathbf{S}_k$ from equation (B3) minimizes J_k.

b)

$$
\begin{aligned}
J_k &= \sum_{i=1}^{Q} \psi_k(\mathbf{x}_i)(y_i - \tilde{y}_i)^2 \\
&= \sum_{i=1}^{Q} \psi_k(\mathbf{x}_i)(y_i - \mathbf{z}_i^T \boldsymbol{\theta}_k (1 - (\mathbf{x}_i - \mathbf{c}_k)^T \mathbf{S}_k (\mathbf{x}_i - \mathbf{c}_k)))^2 \\
&= \sum_{i=1}^{Q} ((\psi_k(\mathbf{x}_i))^{0.5} y_i - (\psi_k(\mathbf{x}_i))^{0.5} \mathbf{z}_i^T \boldsymbol{\theta}_k (1 - (\mathbf{x}_i - \mathbf{c}_k)^T \mathbf{S}_k (\mathbf{x}_i - \mathbf{c}_k)))^2 \\
&= \sum_{i=1}^{Q} ((\psi_k(\mathbf{x}_i))^{0.5} y_i - (\psi_k(\mathbf{x}_i))^{0.5} (1 - (\mathbf{x}_i - \mathbf{c}_k)^T \mathbf{S}_k (\mathbf{x}_i - \mathbf{c}_k)) \mathbf{z}_i^T \boldsymbol{\theta}_k)^2 \\
&= \sum_{i=1}^{Q} ((\breve{y}_i^k - \breve{\mathbf{z}}_i^T \boldsymbol{\theta}_k)^2
\end{aligned}
\quad \text{(B4)}
$$

The problem of computing optimum $\boldsymbol{\theta}_k$ is a least squares optimization problem. Therefore, let $\mathbf{Z}_k$ denote a data matrix whose $i-\text{th}$ row is $\breve{\mathbf{z}}_i^T$ and $\mathbf{Y}_k$ denote an output vector whose $i-\text{th}$ row is $\breve{y}_i^k$. We know from least squares optimization technique that the optimum answer for $\boldsymbol{\theta}_k$ is:

$$
\boldsymbol{\theta}_k = (\mathbf{Z}_k^T \mathbf{Z}_k)^{-1} \mathbf{Z}_k^T \mathbf{Y}_k \quad \text{(B5)}
$$