NOVEL APPROACHES FOR PARAMETER ESTIMATION OF LOCAL LINEAR MODELS FOR DYNAMICAL SYSTEM IDENTIFICATION

Luís Gustavo M. Souza*, Guilherme A. Barreto†

*Federal University of Piauí, Department of Electrical Engineering, Av Ministro Petrônio Portela, S/N
Center of Technology, Campus of Ininga, CEP 64049-550,
Teresina, Piauí, Brazil

†Federal University of Ceará, Department of Teleinformatics Engineering, Av Mister Hull, S/N
Center of Technology, Campus of Pici, CP 6005, CEP 60455-760,
Fortaleza, Ceará, Brazil

Emails: lmotasouza@gmail.com, guialenbar@gmail.com

Abstract— In this paper we introduce two novel techniques for local linear modeling of dynamical system identification. As in standard local model building, we use a vector quantization algorithm, such as the Self-Organizing Map, to partition the joint input-output space into smaller regions. Then, to each neuron we associate a coefficient vector which must be suitably estimated. A comprehensive evaluation of the proposed techniques is carried out for the task of inverse system identification of one benchmarking Single Input/Single Output (SISO) dynamical system. Their performances are compared to those achieved by the Multilayer Perceptron and the Extreme Learning Machine networks. We also evaluate how robust are the proposed techniques with respect to the vector quantization algorithm used to partition the input-output space. The results show that proposed techniques perform better than standard ones for evaluated dataset.

Keywords— Vector quantization, local linear models, nonlinear system identification.

Resumo— Neste trabalho, nós introduzimos duas novas técnicas para modelagem linear local para identificação de sistemas dinâmicos. Como padrão de construção do modelo local, nós usamos um algoritmo de quantização vetorial, tal como o Mapa Auto Organizável, para partição do espaço conjunto de entrada-saída em regiões menores. Então, para cada neurônio nós associamos um vetor de coeficientes que deve ser estimado adequadamente. Uma avaliação abrangente das técnicas propostas é realizada para tarefa de identificação inversa de um sistema dinâmico padrão do tipo Única Entrada/Única Saída. Seus desempenhos são comparados com aqueles realizados pelas redes Perceptron de Múltiplas Camadas e a Máquina de Aprendizagem Extrema. Nós também avaliamos quanto robusto são as técnicas propostas com respeito ao algoritmo de quantização vetorial usado para particionar o espaço de entrada-saída. Os resultados mostram que as técnicas propostas se apresentaram melhores do que as técnicas padrões para o conjuntos de dados avaliado.

Palavras-chave— Quantização vetorial, Modelos lineares locais, Identificação de sistemas não lineares.

1 Introduction

Modern industrial plants have been the source of challenging tasks in dynamical system identification and control (Peng et al., 2007). In particular, the design of control systems to achieve the level of quality demanded by current industry standards requires building accurate models of the plant being controlled.

Since data are usually available in the form of input and output time series, these can be used for building direct and/or inverse models of nonlinear systems by means of computational intelligence methods, such as neural networks (Barreto and Araújo, 2004), Takagi-Sugeno-Kang fuzzy models (Rezaee and Fazel Zarandi, 2010) or hybrid neuro-fuzzy systems (Rubio, 2009), to mention just a few possibilities. Although several techniques for nonlinear dynamical system identification have been proposed (Norgaard et al., 2000), they can be categorized into one of the following approaches: global, local and hybrid models.

In the neural network literature, local modeling techniques have been implemented mostly using the Self-Organizing Map (SOM) to partition the input-ouput space into smaller regions, over which the local models are built (Papadakis and Kaburlasos, 2010; Liu and Djurdjanovic, 2008; Barreto and Souza, 2006). The SOM is an unsupervised competitive learning algorithm which has been commonly applied to clustering, vector quantization and data visualization tasks (Kohonen, 2013). The results reported on those studies are rather appealing, indicating that SOM-based local models can be feasible alternatives to global models based on supervised neural network architectures, such as the Multilayer Perceptron (MLP) and the Extreme Learning Machine (ELM) (Huang et al., 2006).

An important limitation of the aforementioned local linear models is that they were specifically designed to use the SOM algorithm. This means that their performances degrade considerably if another vector quantization (VQ) algorithm is used. The main advantage in using other VQ algorithm than the SOM to partition the input-output space is related to computational costs. There are several VQ algorithms available (Vasuki and Vanathi, 2006), such as the K-Means (Darken and Moody, 1990) and the Frequency-Sensitive Competitive Learning (FSCL) (Ahalt et al., 1990), which are considerably lighter than the SOM and still achieve equivalent data partitioning results.

More recently, a general methodology for building and evaluating local linear models with respect to their robustness (i.e. insensitiveness) to changes in the VQ algorithm has been proposed in Souza and Barreto (2010). This methodology evaluates if the difference between two distributions of residuals generated by the same local model using two different VQ algorithms are statistically significant. For this purpose, the nonparametric Kolmogorov-Smirnov hypothesis test was used. By using this methodology, it was shown that the SOM can be successfully replaced with a lighter VQ algorithm (e.g. the FSCL) without performance loss.

A comprehensive evaluation of the proposed approaches is carried out for the task of inverse system identification of one benchmarking single-input/single-output (SISO) dynamical system. Their performances are compared to those achieved by linear and nonlinear global models, and other local linear modeling techniques available in the literature. In addition, we perform a thorough statistical analysis of the residuals for model validation purposes and also evaluate how robust are the proposed techniques with respect to the VQ algorithm used to partition the input-output space.

The remainder of the paper is organized as follows. In Section 2, the proposed linear local models approaches are then presented. Comprehensive computer simulations and performance analysis of the proposed approaches are presented in Section 3. The paper is concluded in Section 4.

2 The Proposed Approaches

The algorithms to be described in this section are based on the KSOM architecture that was recently proposed as a model to solve system identification problems, such as inverse modeling of real industrial plant (Souza and Barreto, 2008). For training purposes, the KSOM algorithm depends on the VQTAM[1] model (Barreto and Araújo, 2004), which is a simple extension of the SOM algorithm that simultaneously performs vector quantization on the input and output spaces of a given nonlinear mapping.

According to the VQTAM framework, the input vector to the SOM at time step t, $\mathbf{x}(t)$, is composed of two parts. The first part, denoted $\mathbf{x}^{in}(t) \in \mathbb{R}^{p+q}$, carries data about the input of the dynamic mapping to be learned. The second part, denoted $x^{out}(t) \in \mathbb{R}$, contains data concerning the desired output of this mapping. The weight vector of neuron i, $\mathbf{w}_i(t)$, has its dimension increased accordingly. These changes are formulated as follows:

$$\mathbf{x}(t) = \begin{pmatrix} \mathbf{x}^{in}(t) \\ x^{out}(t) \end{pmatrix} \quad \text{and} \quad \mathbf{w}_i(t) = \begin{pmatrix} \mathbf{w}_i^{in}(t) \\ w_i^{out}(t) \end{pmatrix} \tag{1}$$

where $\mathbf{w}_i^{in}(t) \in \mathbb{R}^{p+q}$ and $w_i^{out}(t) \in \mathbb{R}$ are, respectively, the portions of the weight (prototype) vector which store information about the inputs and the outputs of the desired mapping.

Depending on the variables chosen to build the vector $\mathbf{x}^{in}(t)$ and scalar $x^{out}(t)$ one can use the SOM algorithm (or any other VQ algorithm) to learn the forward or the inverse mapping of a given dynamic system. For the inverse identification task we are interested in, we have the following definitions: $\mathbf{x}^{in}(t) = [u(t-1), \ldots, u(t-q); y(t-1), \ldots, y(t-p)]^T$ and $x^{out}(t) = u(t)$ (Norgaard et al., 2000).

The winning neuron at time t is determined based only on $\mathbf{x}^{in}(t)$: $i^*(t) = \arg\min_{\forall i \in \mathcal{A}}\{\|\mathbf{x}^{in}(t) - \mathbf{w}_i^{in}(t)\|\}$. For updating the weights, both $\mathbf{x}^{in}(t)$ and $x^{out}(t)$ are used:

$$\Delta \mathbf{w}_i^{in}(t+1) = \alpha(t) h(i^*, i; t)[\mathbf{x}^{in}(t) - \mathbf{w}_i^{in}(t)], \tag{2}$$
$$\Delta w_i^{out}(t+1) = \alpha(t) h(i^*, i; t)[x^{out}(t) - w_i^{out}(t)], \tag{3}$$

where $0 < \alpha(t) < 1$ is the learning rate, and $h(i^*, i; t)$ is a time-varying Gaussian neighborhood function. Both parameters are calculated similar to the basic SOM algorithm.

2.1 Local Linear Model Based on the K Nearest Prototypes

The first algorithm to be described is called *Prototype-based Multiple KSOM Model* (P-MKSOM), and it was initially proposed in Souza and Barreto (2010). The first step in building the P-MKSOM model requires the VQTAM approach. The P-MKSOM model building starts once VQTAM training is finished.

Firstly, let $j_k^{(i)}$ denote the k-th nearest neighbor of neuron i. Thus, find the K nearest neighbors of the prototype vector $\mathbf{w}_i^{in}$ as follows:

$$\begin{array}{rcl} i_1 & = & \arg\min\limits_{\forall j \neq i} \left\{\|\mathbf{w}_i^{in} - \mathbf{w}_j^{in}\|\right\}, \\ \vdots & \vdots & \vdots \\ i_K & = & \arg\min\limits_{\forall j \neq \{i, i_1, \ldots, i_{K-1}\}} \left\{\|\mathbf{w}_i^{in} - \mathbf{w}_j^{in}\|\right\}, \end{array} \tag{4}$$

where $\mathcal{J}_i = i \cup \{i_k\}_{k=1}^K$ is the set containing the indexes of the K nearest neighbors of the prototype vector $\mathbf{w}_i^{in}$, including neuron i.

Once the set $\mathcal{J}_i$ is determined for each neuron i, we build N local regression models using the prototype vectors whose indexes belong to $\mathcal{J}_i$. Thus, associated to neuron i, we have a coefficient vector $\mathbf{c}_i \in \mathbb{R}^{p+q}$ computed using the least-squares method: $\mathbf{c}_i = \left(\mathbf{R}_i^T \mathbf{R}_i + \lambda \mathbf{I}\right)^{-1} \mathbf{R}_i^T \mathbf{b}_i^{out}$, where $\mathbf{I}$ is

[1] *Vector-Quantized Temporal Associative Memory*

a identity matrix of order $(p+q) \times (p+q)$ and $\lambda > 0$ (e.g. $\lambda = 0.001$) is a small regularization constant. The vector $\mathbf{b}_i^{out} \in \mathbb{R}^{K+1}$ is comprised of the output parts of the K prototype vectors whose indexes belong to $\mathcal{J}_i$, i.e.

$$\mathbf{b}_i^{out} = \begin{bmatrix} w_i^{out} & w_{i_1}^{out} & \cdots & w_{i_K}^{out} \end{bmatrix}^T, \qquad (5)$$

and the matrix $\mathbf{R}_i \in \mathbb{R}^{(K+1)\times(p+q)}$ is comprised of the input parts of the same K prototype vectors:

$$\mathbf{R}_i = \begin{pmatrix} w_{i,1}^{in} & w_{i,2}^{in} & \cdots & w_{i,p+q}^{in} \\ \vdots & \vdots & \vdots & \vdots \\ w_{i_K,1}^{in} & w_{i_K,2}^{in} & \cdots & w_{i_K,p+q}^{in} \end{pmatrix} (6)$$

where the superscript T denotes the transpose vector/matrix.

Once the N local regression models are built, they can be used to approximate the output of the nonlinear mapping of interest. Recall that the P-MKSOM model requires one local model (and hence, one vector of coefficients) per neuron. Which one to use at time t is defined by the index of the winning neuron, $i^*(t)$.

Since we are interested in inverse system identification, the P-MKSOM model estimates the current input $u(t)$ by means of the following equation: $\hat{u}(t) = \mathbf{c}_{i^*}^T \mathbf{x}^{in}(t)$, where the estimation error (residual) at time t is defined as $e(t) = u(t) - \hat{u}(t)$.

2.2 Local Linear Model Estimation Based on the Data Vectors Mapped to the K Nearest Prototypes

The second proposed approach, called *Data-based Multiple KSOM Model* (D-MKSOM), is similar to the P-MKSOM model, differing only in the way the vectors of coefficients $\mathbf{c}_i$, $i = 1, ..., N$, are estimated. Instead of using the prototype vector of neuron i and of its K nearest neighbors, the D-MKSOM model computes the vector of coefficients $\mathbf{c}_i$ of neuron i using the (training) *data vectors* that are mapped to that neuron and to its K nearest neighbors. In other words, in order to estimate the vector $\mathbf{c}_i$, the D-MKSOM model uses all the (training) data vectors belonging to the region formed by the Voronoi cells of neuron i and of its K nearest neighbors.

The first and second steps in building the D-MKSOM model are the same as that for the P-MKSOM: (i) train the VQTAM model using the available training data. (ii) Then, find the set $\mathcal{J}_i = i \cup \{i_k\}_{k=1}^K$ containing the indexes of the K nearest neighbors of the prototype vector $\mathbf{w}_i^{in}$, $i = 1, ..., N$, as defined in (4).

A third step is necessary and consists in finding the set of (training) data vectors that are mapped to the prototypes $\mathbf{w}_i^{in}$, $\mathbf{w}_{i_1}^{in}$, $\mathbf{w}_{i_2}^{in}$, $\ldots$, $\mathbf{w}_{i_K}^{in}$, for $i = 1, \ldots, N$.

Let $n^{(i)}$ be the number of input vectors $\mathbf{x}^{in} \in \mathbb{R}^{p+q}$ mapped to the Voronoi cell of neuron i. Similarly, let $n^{(i_k)}$ be the number of input vectors $\mathbf{x}^{in} \in \mathbb{R}^{p+q}$ mapped to the Voronoi cell of k-th nearest neighbor of neuron i. Hence, the total number of vectors mapped to neuron i and its K nearest neighbors is given by $n_i = n^{(i)} + n^{(i_1)} + n^{(i_2)} + \cdots + n^{(i_K)}$.

Also, let $\mathbf{X}_i^{in}$ be a $(p+q) \times n^{(i)}$ data matrix whose columns are the vectors $\mathbf{x}^{in}$ mapped to the Voronoi cell of neuron i. Finally, let $\mathbf{x}_i^{out} \in \mathbb{R}^{n^{(i)}}$ the vector containing the target outputs x^{out} associated with the vectors $\mathbf{x}^{in} \in \mathbf{X}_i^{in}$.

By the same token, $\mathbf{X}_{i_k}^{in}$ is a $(p+q) \times n^{(i_k)}$ data matrix whose columns are the vectors $\mathbf{x}^{in}$ mapped to the Voronoi cell of neuron i_k, $k = 1, \ldots, K$. Accordingly, $\mathbf{x}_{i_k}^{out} \in \mathbb{R}^{n^{(i_k)}}$ is the vector containing the target outputs x^{out} associated with the vectors $\mathbf{x}^{in} \in \mathbf{X}_{i_k}^{in}$.

Once the pairs $\{\mathbf{X}_i^{in}, \mathbf{x}_i^{out}\}$, $\{\mathbf{X}_{i_1}^{in}, \mathbf{x}_{i_1}^{out}\}$, $\{\mathbf{X}_{i_2}^{in}, \mathbf{x}_{i_2}^{out}\}$, $\ldots$, $\{\mathbf{X}_{i_K}^{in}, \mathbf{x}_{i_K}^{out}\}$, are determined for neuron i and its K nearest neighbors, we can build the local linear model for neuron i.

For this purpose, assuming that the sets $\mathbf{x}_i^{out}$ and $\mathbf{x}_{i_k}^{out}$, $k = 1, \ldots, K$ are arranged as column vectors, we build the vector $\mathbf{b}_i^{out} \in \mathbb{R}^{n_i}$ as follows:

$$\mathbf{b}_i^{out} = \left[\ \mathbf{x}_i^{out}; \mathbf{x}_{i_1}^{out}; \ldots \mathbf{x}_{i_K}^{out}\ \right]_{n_i \times 1}^T. \qquad (7)$$

Similarly, the regression matrix $\mathbf{R}_i \in \mathbb{R}^{n_i \times (p+q)}$ is built using the data matrices $\mathbf{X}_i^{in}$ and $\mathbf{X}_{i_k}^{in}$, $k = 1, \ldots, K$, as follows:

$$\mathbf{R}_i = \begin{pmatrix} \left(\mathbf{X}_i^{in}\right)^T \\ \left(\mathbf{X}_{i_1}^{in}\right)^T \\ \vdots \\ \left(\mathbf{X}_{i_K}^{in}\right)^T \end{pmatrix}_{n_i \times (p+q)}, \qquad (8)$$

where the superscript T denotes the transpose of a vector/matrix.

Hence, the vector of coefficients of neuron i, $\mathbf{c}_i \in \mathbb{R}^{p+q}$, is estimated using the regularized lest-squares method as $\mathbf{c}_i = \left(\mathbf{R}_i^T \mathbf{R}_i + \lambda \mathbf{I}\right)^{-1} \mathbf{R}_i^T \mathbf{b}_i^{out}$, where $\mathbf{I}$ is a identity matrix of dimension $(p+q) \times (p+q)$ and $\lambda > 0$ (e.g. $\lambda = 0.001$) is a small regularization constant.

Once the N local regression models are built, they can be used to approximate the output of the nonlinear mapping of interest. Recall that in this paper we are interested in the inverse identification problem. Thus, the D-MKSOM model estimates $u(t)$ by means of the following equation: $\hat{u}(t) = \mathbf{c}_{i^*}^T \mathbf{x}^{in}(t)$, where the estimation error (residual) at time t is defined as $e(t) = u(t) - \hat{u}(t)$.

3 Computer Simulations and Discussion

The performances of the proposed SOM-based local linear models are compared to MLP- and ELM-based global models in the task of inverse system identification. We also compare the proposed models with other local modeling approaches, such as the VQTAM, KSOM and the

Local Linear Mapping (LLM) (Walter et al., 1990). The LLM associates a linear model to each neuron in the SOM and estimates their vectors of coefficients using a variant of the least mean squares (LMS) adaptation rule. All simulations were carried out in Matlab®.

All the models are initially evaluated via the statistics of the normalized mean-squared estimation error (NMSE) computed for the testing time series: $NMSE = \sum_{t=1}^{M} e^2(t)/M \cdot \hat{\sigma}_u^2 = \hat{\sigma}_e^2/\hat{\sigma}_u^2$, where $\hat{\sigma}_u^2$ is the variance of the original time series $\{u(t)\}_{t=1}^{M}$ and M is the length of the sequence of residuals for the testing time series.

Additionally, a residual analysis is presented for the best models. Finally, hypothesis testing is carried out to analyze the influence of the VQ algorithm on the performance of the local models. Each hypothesis testing is implemented through the Kolmogorov-Smirnov test (Soong, 2004) on the estimation error distribution generated by a given model.

The heat exchanger data set comes from a liquid-satured steam heat exchanger (Bittanti and Piroddi, 1997), where water is heated by pressurized saturated steam through a copper tube. The motivation for the choice of the heat exchanger as a benchmark is that this plant is characterized by a non-minimum phase behavior which makes the design of controllers particulary challenging even in a linear context. The measured values of the liquid flow rate (in m^3/s) defines the input time series, while the outlet liquid temperature (in Celsius degrees) defines the output time series. The sampling rate was set to 1s.

Results on the NMSE criterion: We included performance results of variants of the P-MKSOM and D-MKSOM models obtained by replacing the SOM with other VQ algorithms in the VQTAM model. All the models were trained using the first 2200 samples of the input/output time series, validated with the following 1000 samples and tested with the remaining 800 samples. Input/output time series were rescaled to the $[-1, +1]$ range.

The best configuration found for the MLP-1h and MLP-LM models has 20 hidden neurons (100 independent training/validation runs for each value of N in the range of 2 to 50). The number of neurons of the first hidden layer of the MLP-2h model was then set to 20, while for the second hidden layer it was set to half the number of neurons of the first hidden layer. The MLP-1h and MLP-LM were trained with constant learning rate equal to 0.1. For the ELM model, the number of hidden neurons was set to 20, same value used by the MLP-1h and MLP-LM models. The memory orders were set to $p = 6$ and $q = 3$, respectively.

Table 1: Performance results for the heat exchanger data.

Neural Models	NMSE			
	mean	*min*	*max*	*std*
ELM	0.3346	0.3316	0.3391	1.50e-03
D-MKSOM	0.3798	0.3657	0.3918	4.63e-03
MLP-1h	0.4292	0.4173	0.4501	6.34e-03
P-MKSOM	0.5640	0.4137	0.8118	0.1044
MLP-LM	0.5672	0.2140	0.6016	0.0692
KSOM	0.5841	0.4139	1.3877	0.1431
VQTAM-T	0.7701	0.4774	1.2345	0.1330
LLM	0.7837	0.7378	0.8702	0.0242
Linear	0.9257	0.9257	0.9257	2.30e-07
VQTAM-G	1.2396	1.1828	1.3129	0.0253
MLP-2h	1.3003	1.2207	1.5466	0.0768

Table 2: Performance results for the P-MKSOM and D-MKSOM models using different VQ algorithms (heat exchanger data).

P-MKSOM Method				
VQ Algorithms	*NMSE*			
	mean	*min*	*max*	*std*
SOM	0.5640	0.4137	0.8118	0.1044
WTA	0.6026	0.5382	0.6713	0.0316
K-means	0.6744	0.6503	0.6758	4.35e-03
FCL	1.4722	1.3263	1.7196	0.0831
FSCL	1.5817	1.2887	2.4085	0.3259

D-MKSOM Method				
VQ Algorithms	*NMSE*			
	mean	*min*	*max*	*std*
SOM	0.3798	0.3657	0.3918	4.64e-03
FCL	0.4054	0.3770	0.4311	0.0112
WTA	0.4066	0.3941	0.4423	9.22e-03
FSCL	0.4082	0.3969	0.4304	7.79e-03
K-means	0.4083	0.4001	0.4170	3.42e-03

The best number of neurons found for the VQTAM model after experimentation on the validation set was $N = 30$ (100 independent training/validation runs for each value of N ranging from 5 to 50). This number of neurons was then used by all the other SOM-based models. The initial and final learning rates were set to $\alpha_0 = 0.5$ and $\alpha_T = 0.001$. The initial and final values of the neighborhood function radius are $\sigma_0 = N/2$ and $\sigma_T = 0.001$. The learning rate for the LMS part of the LLM model was set to 0.1. The optimal number of nearest neighbors for the KSOM was found to be around $K = 20$ (100 independent training/validation runs for each value of K ranging from 1 to 30). This value was then used by the P-MKSOM and D-MKSOM models. The obtained results are shown in Table 1.

This time, the best performance was achieved by the ELM global model. Note, however, the performance of the D-MKSOM model is better than the MLP-1h model and it is comparable to that of the ELM model. Again, the VQTAM with topological interpolation performed better than the one with geometric interpolation. It is worth noting that, this time, among the five best ranked models, three of them are global nonlinear models.

The influence of the VQ algorithm on the performances of the P-MKSOM and D-MKSOM approaches is analyzed in Table 2 for the heat exchanger dataset. The local models in question are implemented using the following VQ algorithms: K-means, WTA, FSCL and FCL. By analyzing this table we can observe that, for both the P-MKSOM and D-MKSOM methods, the best local models were the ones generated by the SOM, an possible indication that topology preservation is important for the proposed models.

Residual Analysis: It is an important tool for model validation, since it allows the user to assess how well the model learns the dynamics of the training data and how well it respects the modeling assumptions. The most common modeling assumption is that noise resembles a Gaussian white noise process (Norgaard et al., 2000). By analyzing the sequence of residuals produced by each model (local and global ones) using the testing data, the user can assess the degree of matching between the statistical properties of the sequence of residuals and the theoretic modeling assumptions.

For nonlinear models, the statistical tests used for validating of the identified model are listed below (Billings and Zhu, 1994):

$$\Phi_{ee}(\tau) = E\{e(t-\tau)e(t)\} = \delta(\tau) \quad (9)$$

$$\Phi_{ue}(\tau) = E\{u(t-\tau)e(t)\} = 0, \forall\tau \quad (10)$$

$$\Phi_{u^{2\prime}e}(\tau) = E\{[u^2(t) - \overline{u^2(t)}]e(t-\tau)\} = 0, \forall\tau \quad (11)$$

$$\Phi_{u^{2\prime}e^2}(\tau) = E\{[u^2(t) - \overline{u^2(t)}]e^2(t-\tau)\} = 0, \forall\tau \quad (12)$$

where $E\{\cdot\}$ denotes the expected value operator, $e(t)$ denotes the residual (error) obtained by the model at time t using testing data, $u(t)$ is the corresponding input sample, and $\delta(\tau)$ is the Kronecker delta function. The overbar denotes the time average operation. The prime symbol ($'$) in Equations (11)-(12) denotes that the mean level has been removed from the corresponding data sequence.

The results on the estimation of the linear and nonlinear ACF's/XCF's are shown in Figure 1. We shown only the results for the D-MKSOM, ELM and P-MKSOM models, since they presented the best performances for this data set. By analyzing these figures, one can note that the assumptions of gaussianity and uncorrelatedness of the sequence of residuals are mostly satisfied by the three models in these data sets.

Robustness Analysis of the Proposed Models: The final set of experiments aims at evaluating the degree of similarity, from a statistical viewpoint, among the sequence of residuals generated by the P-MKSOM and D-MKSOM methods for different VQ algorithms. For this purpose, we use the Kolmogorov-Smirnov test (KS-test) (Soong, 2004). The KS-test quantifies a distance between the empirical cumulative distribution functions (CDF) of two sequences of residuals. The null hypothesis to be tested is that the sequences are drawn from the same distribution.

Table 3: KS-test results on the P-MKSOM and D-MKSOM performances.

P-MKSOM		**D-MKSOM**	
VQ Algorithms	**KS-test Results**	**VQ Algorithms**	**KS-test Results**
FCL	Reject	FCL	Accept
FSCL	Accept	FSCL	Accept
WTA	Accept	WTA	Accept
K-means	Accept	K-means	Accept

Table 3 presents the results for the P-MKSOM and D-MKSOM models. A rejection of the null hypothesis indicates that the CDF of the residuals generated by the original MKSOM model[2] is different from the CDF of residuals generated by the MKSOM implemented with a different VQ algorithm. The acceptance of the null hypothesis indicates that the CDF of the residuals generated by the original MKSOM models is equivalent to the CDF of residuals generated by the MKSOM models implemented with a different VQ algorithm.

From Table 3 one can infer that, the performance of the original P-MKSOM model is statistically equivalent to those obtained by implementing the P-MKSOM with the FSCL, WTA and K-means algorithms. In case of equivalence, we recommend the user to choose the computationally lighter VQ algorithm to build the linear local model of interest (the WTA algorithm, in this case). On the other hand, the performance of the original D-MKSOM model is statistically equivalent to those obtained by implementing the D-MKSOM with all algorithms. As mentioned before, in case of equivalence, we recommend the user to choose the computationally lighter VQ algorithm.

4 Conclusions

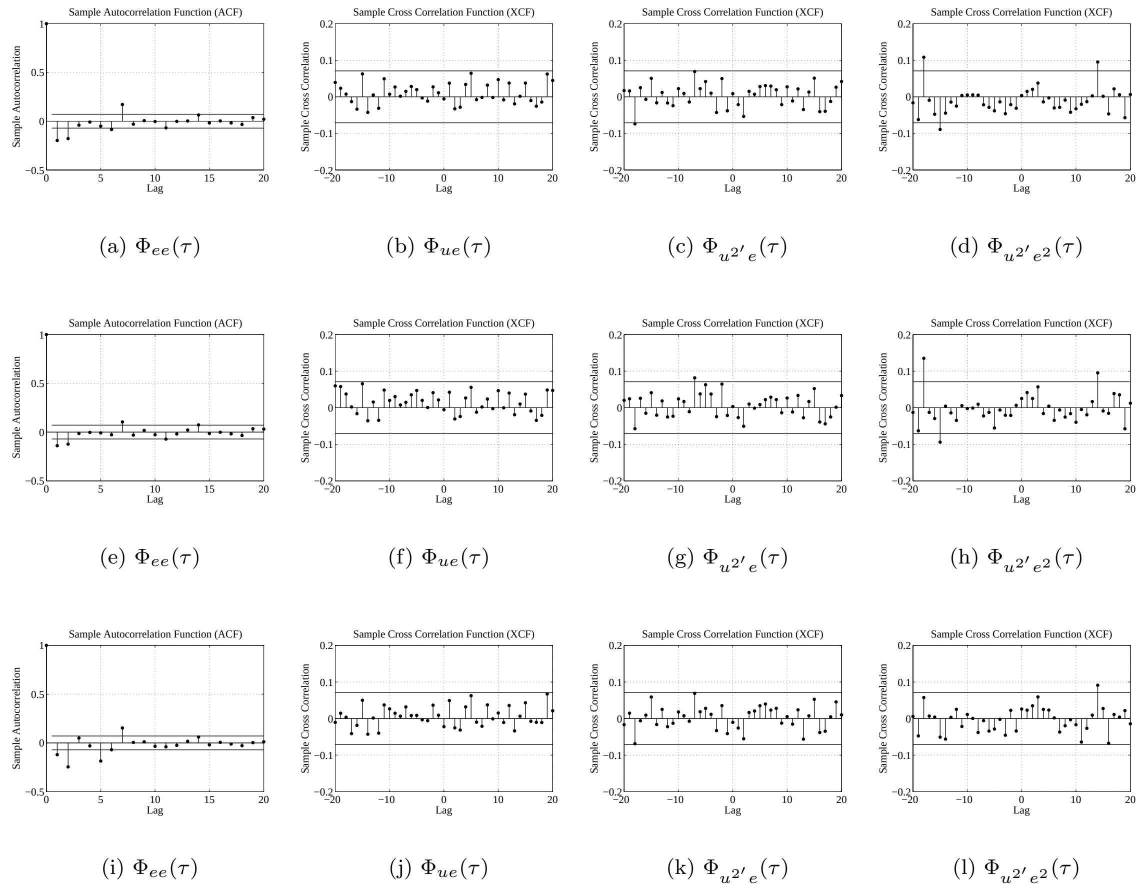
From the exposed in this paper, the main conclusion is that the proposed local linear models (D-MKSOM and P-MKSOM) consistently outperformed standard global models for system identification based on MLP and ELM neural networks, mainly the D-MKSOM model. We also verified that the D-MKSOM model is more robust that the P-MKSOM model to changes in the base vector quantization algorithm.

References

Ahalt, S., Krishnamurthy, A., Cheen, P. and Melton, D. (1990). Competitive learning algorthms for vector quantization, *Neural Networks* **3**: 277–290.

Barreto, G. A. and Araújo, A. F. R. (2004). Identification and control of dynamical systems using the self-

[2] By the original MKSOM models, we mean the P- and D-MKSOM models that use the VQTAM method with the SOM as the VQ algorithm.



(a) $\Phi_{ee}(\tau)$ (b) $\Phi_{ue}(\tau)$ (c) $\Phi_{u^{2'}e}(\tau)$ (d) $\Phi_{u^{2'}e^2}(\tau)$

(e) $\Phi_{ee}(\tau)$ (f) $\Phi_{ue}(\tau)$ (g) $\Phi_{u^{2'}e}(\tau)$ (h) $\Phi_{u^{2'}e^2}(\tau)$

(i) $\Phi_{ee}(\tau)$ (j) $\Phi_{ue}(\tau)$ (k) $\Phi_{u^{2'}e}(\tau)$ (l) $\Phi_{u^{2'}e^2}(\tau)$

Figure 1: Residual correlation analysis for the heat exchanger data: (a,b,c,d) D-MKSOM model, (e,f,g,h) ELM model and (i,j,k,l) P-MKSOM model.

organizing map, *IEEE Transactions on Neural Networks* **15**(5): 1244–1259.

Barreto, G. A. and Souza, L. G. M. (2006). Adaptive filtering with the self-organizing maps: A performance comparison, *Neural Networks* **19**(6): 785–798.

Billings, S. A. and Zhu, Q. M. (1994). Nonlinear model validation using correlation tests, *International Journal of Control* **60**(6): 1107–1120.

Bittanti, S. and Piroddi, L. (1997). Nonlinear identification and control of a heat exchanger: A neural network approach, *Journal of the Franklin Institute* **334**(1): 135–153.

Darken, C. and Moody, J. (1990). Fast adaptive k-means clustering: Some empirical results, *Proceedings of the International Joint Conference on Neural Networks (IJCNN'90)*, Vol. 2, pp. 233–238.

Huang, G. B., Zhu, Q. Y. and Ziew, C. K. (2006). Extreme learning machine: Theory and applications, *Neurocomputing* **70**(1–3): 489–501.

Kohonen, T. (2013). Essentials of the self-organizing map, *Neural Networks* **37**: 52–65.

Liu, J. and Djurdjanovic, D. (2008). Topology preservation and cooperative learning in identification of multiple model systems, *IEEE Transactions on Neural Networks* **19**(12): 2065–2072.

Norgaard, M., Ravn, O., Poulsen, N. K. and Hansen, L. K. (2000). *Neural Networks for Modelling and Control of Dynamic Systems*, Springer-Verlag.

Papadakis, S. E. and Kaburlasos, V. G. (2010). Piecewise-linear approximation of non-linear models based on probabilistically/possibilistically interpreted intervals numbers (INs), *Information Sciences* **180**(24): 5060–5076.

Peng, H., Nakano, K. and Shioya, H. (2007). A comprehensive review for industrial applicability of artificial neural networks, *IEEE Transactions on Control Systems Technology* **15**(1): 130–143.

Rezaee, B. and Fazel Zarandi, M. (2010). Data-driven fuzzy modeling for Takagi-Sugeno-Kang fuzzy system, *Information Sciences* **180**(2): 241–255.

Rubio, J. J. (2009). SOFMLS: Online self-organizing fuzzy modified least-squares network, *IEEE Transactions on Fuzzy Systems* **17**(6): 1296–1309.

Soong, T. T. (2004). *Fundamentals of Probability and Statistics for Engineers*, 1st edn, John Wiley & Sons, West Sussex, England.

Souza, L. G. M. and Barreto, G. A. (2008). Multiple local ARX modeling for system identification using the self-organizing map, *Proceedings of II European Symposium on Time Series Prediction (ESTSP'08)*, pp. 215–224.

Souza, L. G. M. and Barreto, G. A. (2010). On building local models for inverse system identification with vector quantization algorithms, *Neurocomputing* **73**(10-12): 1993–2005.

Vasuki, A. and Vanathi, P. T. (2006). A review of vector quantization techniques, *IEEE Potentials* **25**(4): 39–47.

Walter, J., Ritter, H. and Schulten, K. (1990). Non-linear prediction with self-organizing map, *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN'90)*, Vol. 1, pp. 587–592.