BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 61, No. 3, 2013
DOI: 10.2478/bpasts-2013-0071

Modified neuro-fuzzy TSK network and its application in electronic nose

S. OSOWSKI[1,2*], K. BRUDZEWSKI[1], and L. TRAN HOAI[3]

[1] Institute of Theory of Electrical Engineering, Warsaw University of Technology, 75 Koszykowa St., 00-662 Warsaw, Poland
[2] Military University of Technology, 2 S. Kaliskiego St., 00-908 Warsaw, Poland
[3] Hanoi University of Technology, 1 Dai Co Viet Road, Hanoi, Vietnam

Abstract. The paper develops the modified structure of the Takagi-Sugeno-Kang neuro-fuzzy network with a theoretical basis for its adaptation. The simplified structure follows from the basic theoretical considerations concerning the way of creating the inference rules. The important point of this solution is the application of the fuzzy clustering algorithm to the input data. The efficiency of the proposed solution has been checked on the examples of regression and classification problems concerning the electronic nose.

Key words: neuro-fuzzy TSK networks, fuzzy clusterization, regression, classification.

1. Introduction

The neuro-fuzzy networks are regarded as a very useful tool for solving many practical problems concerning the processes of fuzzy behavior. The fuzzy mathematical relations forming the basis of this approach are capable to reflect such processes in much better way than a classical crisp approach. One of the most often exploited neuro-fuzzy structure is the adaptive neuro-fuzzy inference system (ANFIS), directly implementing the Takagi – Sugeno – Kang (TSK) linear inference rule [1–3]. The advantage of this solution is an easy adaptation algorithm for the system parameters, relying directly their values on the learning data set. Thanks to the strict division between the nonlinear parameters of the fuzzy rules and the linear parameters of the output layer we can split the learning algorithm into two stages: the optimization of the nonlinear parameters of the fuzzification functions and then adaptation of the linear output weights of the network.

The most important problem in the design of the network is the creation of inference rules [4–6]. The standard ANFIS TSK system, while operating on the combination of the membership functions for each variable, is inefficient at many inputs, since it results in an extremely large number of learning rules, exploding with the increase of input nodes [1]. Moreover, most of these rules are empty while operating in regions free of data.

In this paper we propose to solve this problem by applying the fuzzy clusterization of the input data and associating each cluster with one independent inference rule. The center of each cluster is automatically the center of the premise part of the rule. In the clusterization process we apply the very efficient Gustafson-Kessel (GK) self-organizing algorithm, where the number of clusters is generated automatically using specially adopted measures of quality [4].

It is well known that the TSK neuro-fuzzy inference system can serve as a universal data approximation system arbitrary accuracy [4–8]. We illustrate this property by two practical examples concerning the artificial nose application. The first example represents the solution of the regression problem and the second one – of the classification problem.

2. Neuro-fuzzy TSK approximation

Let us consider a multi-dimensional process of N inputs, given by the vector $\mathbf{x}$, and one output node, to be modeled as a fuzzy system of Takagi-Sugeno-Kang type [8]. This system implements the inference rules, that can be stated in general vector form as follows

$$\textit{if}\ \mathbf{x}\ \textit{is}\ \mathbf{A}\ \textit{then}\ y = f(\mathbf{x}), \tag{1}$$

where $\mathbf{x} = [x_1, x_2, ..., x_N]^T$ and $f(\mathbf{x})$ is the crisp linear function of the form

$$f(\mathbf{x}) = p_0 + p_1 x_1 + p_2 x_2 + ... + p_N x_N. \tag{2}$$

The TSK coefficients p_i $(i = 0, 1, \ldots, N)$ of this function are adjusted automatically in the learning process. The premise *if* $\mathbf{x}$ *is* $\mathbf{A}$ represents the fuzzifier, which is usually defined for each variable x_i as the Gaussian, a bell function or other function [7]. Fuzzy inference system usually employ many rules of the form (1). Let us denote the firing level of each inference rule at the input, represented in the form of vector $\mathbf{x}$, by the weight μ_i and its output by $y_i(\mathbf{x})$. Then, with L inference rules, the output of the fuzzy system at the excitation given by the vector $\mathbf{x}$ may be described in the form

$$y(\mathbf{x}) = \frac{\sum\limits_{i=1}^{L} \mu_i y_i(\mathbf{x})}{\sum\limits_{j=1}^{L} \mu_j} = \frac{\sum\limits_{i=1}^{L} \mu_i \left(p_{i0} + \sum\limits_{j=1}^{N} p_{ij} x_j \right)}{\sum\limits_{j=1}^{L} \mu_j}. \tag{3}$$

[*] e-mail: sto@iem.pw.edu.pl

This rule is known as the universal approximation formula [4, 6]. Let us replace the original parameters p_{ij} of TSK functions by their relative values defined

$$p_{ij} := \frac{p_{ij}}{\sum_{j=1}^{L} \mu_j}. \tag{4}$$

Equation (3) describing the response of the fuzzy system to the excitation $\mathbf{x}$ may be now rewritten in the form

$$y_i(\mathbf{x}) = \sum_{i=1}^{L} \mu_i \left(p_{i0} + \sum_{j=1}^{N} p_{ij} x_j \right). \tag{5}$$

This equation may be associated with the neuro-fuzzy network structure presented in Fig. 1. Each horizontal channel represents one inference rule composed of the fuzzification block $\mu_i(\mathbf{x})$ and the generator of TSK linear function $y_i = f_i(\mathbf{x})$.

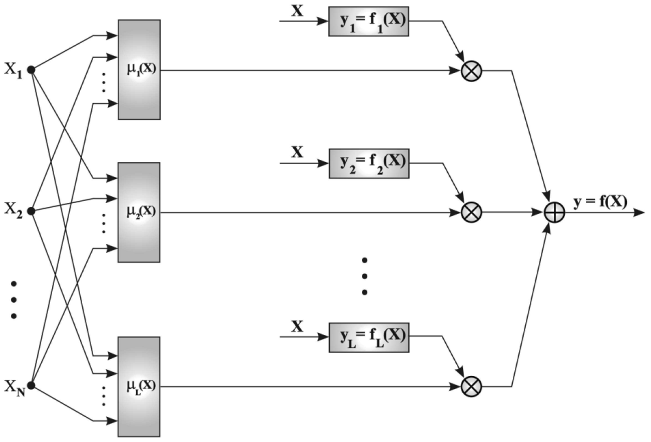

Fig. 1. The neuro-fuzzy network structure corresponding to the modified ANFIS TSK system described by Eq. (5)

If we assume that the values of μ_i are known (earlier determined) for the particular excitation vector $\mathbf{x}$, the last relation represents a linear equation of the unknown parameters p_{ij}. Determination of their values is then straightforward possible and is simplified to the solution of the set of linear equations written for all learning data ($\mathbf{x}_i$, $y(\mathbf{x}_i) = d_i$), for $i = 1, 2, \ldots, p$, where p is the number of learning data pairs. With L inference rules and N inputs of the network the number of unknown parameters p_{ij} is $L(N+1)$. It means that for p learning data pairs we have to solve the system of p equations with $L(N+1)$ unknown parameters, where $p > L(N+1)$. This can be effectively done by applying SVD decomposition and a pseudo-inverse technique.

In this way the most important problem of designing the fuzzy system has been transferred to the generation of optimal inference rules. Inference rules in classical ANFIS approach are created as the combinations of the membership functions for each input variable [1, 9, 10]. However this method leads to an exponential increase of the rules at increasing number of the input variables. Moreover, a lot of these rules are empty, since it is impossible to avoid the operation of the system in the regions deprived of data for some specific combinations of membership functions of different input variables.

3. Generation of inference rules using fuzzy self-organization of data

A better way of inference rule generation is to localize them in the regions where sufficient input data exist. This is made possible by an initial data clusterization step (e.g. using a self-organization approach). Then the inference rules will be associated with the clusters and the centers of the rules should be placed in the central points of these clusters. Hence, each inference rule will operate in the active region, taking part in total representation of the input space. The best way of penetrating the data space is the mechanism of fuzzy self-organization.

There are many different fuzzy self-organization solutions. To the most often used belong the algorithms of: c-means [11], Gustafson-Kessel [12] or Gath-Geva [13]. In our solution we apply the Gustafson-Kessel modification of c-means. The value $\mu_i(\mathbf{x}_j)$ represents the membership function of the j-th input vector $\mathbf{x}_j$ to the i-th cluster of the center $\mathbf{c}_i$ and is described by [11, 14]

$$\mu_i(\mathbf{x}_j) = \frac{1}{\sum_{k=1}^{L} \left(\frac{d(\mathbf{c}_i, \mathbf{x}_j)}{d(\mathbf{c}_k, \mathbf{x}_j)} \right)^{2/(m-1)}}. \tag{6}$$

In this relation L is the number of clusters and m – the parameter controlling the fuzziness of the cluster (typical value $m = 2$). The function $d(\mathbf{c}_i, \mathbf{x}_j)$ represents the scaled distance between the vectors $\mathbf{x}_j$ and $\mathbf{c}_i$. This distance in the Gustafson-Kessel algorithm is described by the relation [12]

$$d^2(\mathbf{c}_i, \mathbf{x}_j) = (\mathbf{x}_j - \mathbf{c}_i)^T \sqrt[N]{\det(\mathbf{F}_i)}\mathbf{F}_i^{-1} (\mathbf{x}_j - \mathbf{c}_i) \tag{7}$$

in which $\mathbf{F}_i$ is the i-th cluster covariance matrix, defined in the way

$$\mathbf{F}_i = \frac{\sum_{j=1}^{p} \mu_i^m(\mathbf{x}_j) (\mathbf{x}_j - \mathbf{c}_i) (\mathbf{x}_j - \mathbf{c}_i)^T}{\sum_{j=1}^{p} \mu_i^m(\mathbf{x}_j)}. \tag{8}$$

At random initialization of centers they are successively updated using the typical rule of $\mathbf{c}$-means

$$\mathbf{c}_i = \frac{\sum_{j=1}^{p} \mu_i^m(\mathbf{x}_j)\mathbf{x}_j}{\sum_{j=1}^{p} \mu_i^m(\mathbf{x}_j)}. \tag{9}$$

The iterations are continued until the objective function $E = \sum_{i=1}^{L} \sum_{j=1}^{p} \mu_i^m(\mathbf{x}_j) d^2(\mathbf{c}_i, \mathbf{x}_j)$ is minimized. Such an organization of clustering provides a perfect adjustment of clusters to the shape of data clouds they represent. Moreover, the clusters correspond only to the regions, where the data exist.

The next problem is the determination of the optimal number of clusters representing the inference rules. More rules mean better representation of data, but at the same time – an increase of the complexity of the network and potential loss of generalization properties leading to the deterioration of the quality of the network operation in the testing mode.

To control the number of clusters we apply here the heuristic validity measures [4] that assess different qualities of the clusters. On the basis of these measures the sub-optimal number of clusters can be explicitly determined. We have applied four different validity measures [4, 14]

- The fuzzy hypervolume measure V_h

$$V_h = \sum_{i=1}^{L} \sqrt{\det(\mathbf{F}_i)}; \tag{10}$$

- The average partition density D_A

$$D_A = \frac{1}{L} \sum_{i=1}^{L} \frac{S_i}{\sqrt{\det(\mathbf{F}_i)}}, \tag{11}$$

  where the parameter $S_i = \sum\limits_k \mu_i(\mathbf{x}_k)$ is calculated for all vectors $\mathbf{x}_k$ satisfying the basic relation $(\mathbf{x}_k - \mathbf{c}_i)^T \mathbf{F}_i^{-1} (\mathbf{x}_j + \mathbf{c}_i) < 1$;
- The average within cluster distance D_w

$$D_w = \frac{1}{L} \sum_{i=1}^{L} \frac{\sum\limits_{k=1}^{p} \mu_i^m(\mathbf{x}_k) d(\mathbf{c}_i, \mathbf{x}_k)}{\sum\limits_{k=1}^{p} \mu_i^m(\mathbf{x}_k)}; \tag{12}$$

- The average cluster flatness t_A

$$t_A = \frac{1}{L \sum\limits_{i=1}^{L} t_i}. \tag{13}$$

In the last equation $t_i = \lambda_{i\,\min}/\lambda_{i\,\max}$ is the ratio between the smallest and the largest eigenvalue of the i-th cluster covariance matrix $\mathbf{F}_i$. A good partition of data is indicated by small values of V_h and t_A and high values of D_A and D_w [4]. To find the compromise between all these requirements we define the global quality factor q

$$q = a_1 V_h - a_2 D_A - a_3 D_w + a_4 t_A \tag{14}$$

which is subject to minimization. The coefficients a_1, a_2, a_3 and a_4 are positive scaling weights of their sum equal to one. Their values are adjusted by trying different proportions among them and taking the values leading to the best results of estimation on the validation data, or applying the optimization approach. The number of clusters corresponding to the minimum of this measure indicates the optimal or suboptimal number of inference rules [4]. In determining this value we have to try different number of clusters, partition the data into these clusters, calculate the values of q and select the number which corresponds to the smallest value of q. If there are few minima we should *try all of them in further signal processing*. The procedure of designing the TSK structure may be now presented in the following form.

- First we perform the Gustafson-Kessel clusterization of the input data described by the set of p available input vectors $\mathbf{x}_j$ $(j = 1, 2, \ldots, p)$. This procedure includes the determination of the optimal number of clusters.
- After determining the clusters we associate the firing level $\mu_i(\mathbf{x}_j)$ of each inference rule for the vector $\mathbf{x}_j$ with the membership value of this vector to the particular i-th cluster by using the relation (6). In this way we fix the values of μ_i in the equation (5).
- In the next step we form the set of linear equations (5) for data pairs $(\mathbf{x}_k, d_k)$:

$$d_k = \sum_{i=1}^{L} \mu_i \left(p_{i0} + \sum_{j=1}^{N} p_{ij} x_{jk} \right) \tag{15}$$

  for $k = 1, 2, \ldots, p$, where x_{jk} is j-th entry of k-th input vector $\mathbf{x}_k$. The solution of this set of linear equations (for example in the form of pseudoinverse) results in the optimal parameter values p_{ij} of TSK functions of all inference rules.

4. Application of neuro-fuzzy network in electronic nose

In recent years the gas sensors organized in the form of electronic nose have been widely used in many practical applications, including such distant areas as cosmetic, food industry or military purposes (detecting explosive materials by measuring the smells using gas sensors). To counteract poor selectivity of sensors we use, in practice, the array of sensors (the heart of the electronic nose) containing several sensors with different cross sensitivities. The application of a sensor array enables to characterize the gas mixture by many sensor signals, formed by the weighted sum of sensor reaction signals to the gas components.

Such sensor elements need calibration to assign their signals with the actual concentration of each gas component in the measured mixture. The signals of the gas sensors used in an array after some normalization form the input, applied to the nonlinear calibration network. The aim of calibration network is to develop the signals that are proportional to the concentration of each gas forming the mixture. This kind of task belongs to the regression problems.

The other problem is to recognize only the presence or absence of some gas in the environment under measurement, or the recognition of aroma, that is the composition of some gases. This is the gas detection and classification task, very common in food, cosmetic or petrol industry. We will show now that the neuro-fuzzy network developed in the previous section is well suited for both types of tasks performed by an electronic nose.

The particular problem, we consider in this work, is the measurement of the gaseous mixture of four gases: carbon oxide, methane, methanol and propane/butane [15]. The measurements have been done by using 5 semiconductor sensors: three of Figaro type and two Nemoto sensors, placed in the test chamber through which the laminar gas flow of controlled temperature (23°C) and humidity (RH = 70%) was forced. The sensor signals are delivered to a PC computer by module Rev.D1 type ADAM-4017 (with a 8-channel analog input) for further processing. To stabilize the results of sensor signals we take into account the averaged temporal series of the sensor resistances $R(j)$, one for each j-th sensor of the array. In order to produce the consistent data we have to reduce the effect of the baseline. To achieve this we rely on the differential signals and use the relative variations $r(j)$ for all sensor resistances ($j = 1, 2, ..., 5$)

$$r(j) = \frac{R(j) - R_o(j)}{R_o(j)}, \tag{16}$$

where $R(j)$ is the actual resistance of the j-th sensor in the array and $R_o(j)$ represents the baseline value of the resistance. As the reference, the baseline value of the measured resistance of each sensor in the synthetic air atmosphere is used. They form the feature vector $\mathbf{x} = [r(1), r(2), ..., r(5)]^T$. The feature vector created in this way should be associated with the target values representing the concentrations of four investigated gases. They form the target vector **d**. In real applications the target values may be either the actual concentrations of the gases (regression problem) or the fact of presence or absence of the particular gas in the gaseous mixture (the classification problem).

Different concentrations of gases have been used in experiments. The maximum level of concentration did not exceed 1000 ppm and the smallest one was zero (lack of the gas). The number of different measured gas mixtures tested in our numerical experiments was 420. Figure 2 presents the change of the concentrations of four gases in these 420 measurements.

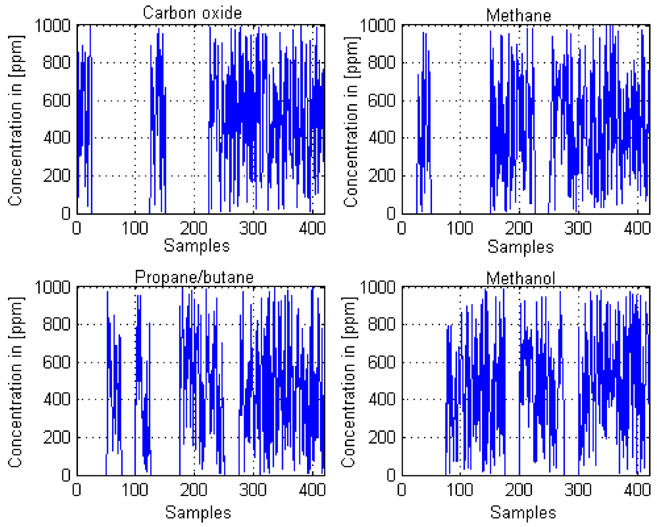

Fig. 2. The concentrations of four gases used in the experiments

The choice of sensors for an electronic nose device should guarantee to observe their different reactions to the gases. In our experiments we have applied 3 Figaro series sensors (TGS-815, TGS-822 and TGS-842) and two of Nemoto type (NAP-11A and NAP-11AE). To find the dependence between the sensor signals we have calculated the normalized correlation coefficients between pairs of signals in 420 measurements. The results are presented in Table 1 in the form of a correlation matrix. As it is seen there is a low correlation amount between the signals generated by the sensors. It verifies that such set of sensors should have sufficient selectivity if applied to the gas components.

Table 1
The relative correlation coefficient matrix between the signals of all 5 sensors applied in the electronic nose

	s_1	s_2	s_3	s_4	s_5
s_1	1.0000	0.1260	0.0844	0.2088	0.5594
s_2	0.1260	1.0000	0.1978	0.1104	0.5640
s_3	0.0844	0.1978	1.0000	0.0595	0.5410
s_4	0.2088	0.1104	0.0595	1.0000	0.5275
s_5	0.5594	0.5640	0.5410	0.5275	1.0000

In solving the nose estimation problem we have applied the neuro-fuzzy network working in two different modes. The first one is the classification task, in which the system should recognize only the absence or presence of each component in the gaseous mixture. It means formation of a four-classes system by rules having the same premise part and differing only by the consequent parts. We have applied the approach presented in Sec. 2 of this paper.

The first step is to make data clustering and determine the optimal number of clusters (inference rules) by using the heuristic validity measures defined by the relations (10)–(14).

Figure 3 illustrates the change of the global quality factor q as a function of different number of applied rules. The curve is non-monotonic and possesses four minima: at $L = 3, 5, 8$ and 12. To find which of them is the best one in our application we have performed the learning and testing the TSK network at all of them. We have also included the case of 6 rules, the number very close to 5, but at which the quality factor achieved value far from optimum.

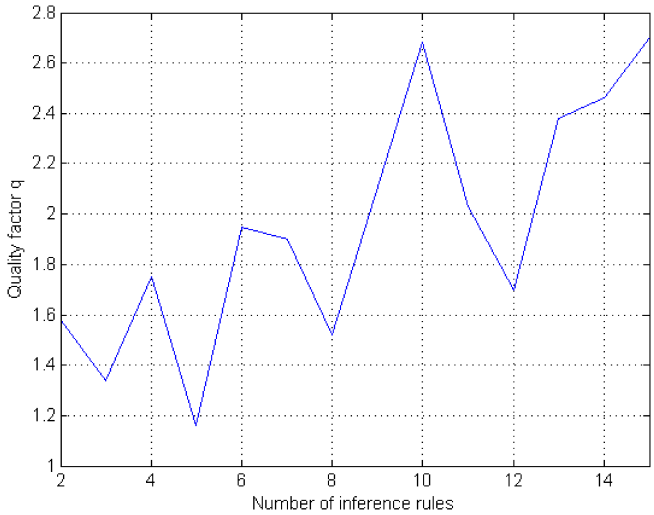

Fig. 3. The change of quality factor q at different number of inference rules in gas measurement problem

In these experiments we have split the data into two parts: 220 randomly chosen samples were used for learning and the remaining 200 samples – for testing only. The original data have been normalized by dividing all of them by the maximum value of the sensor signals. The TSK network contained four outputs, each representing the particular gas. For each input vector **x** the particular outputs were coded as one, when the appropriate gas was present and by zero, when it was absent in the mixture. For example, at the mixture containing only the carbon oxide and methane the code of the target vector was **d** = [1 1 0 0] with the first two ones representing gases present in the mixture and the remaining zeros responsible for the absent gases (propane/butane and methanol). The actual results of recognition are generated by the TSK network as the real numbers forming vector **y**. These numbers are then rounded to zero (absence of the particular gas) when the appropriate absolute value was smaller than 0.5 and in other case it was set to 1 (presence of the gas).

The sample under investigation was counted as properly classified when the presence or absence of all gases were correctly recognized. If at least one of the gases was recognized falsely it was treated as misclassification. All learning stages have been performed using Matlab [16]. After learning stage the parameters of the network have been fixed and the real testing, applying the data not used in the learning stage has been done. We have noted separately the classification results of learning-set and testing-set samples.

It is evident that by increasing the number of inference rules we should get better classification results in the learning mode. However, it is not necessarily true for the testing mode data. In Table 2 we depict the results of learning and testing the data in learning and testing modes. We present the number of misclassifications as well as its percentage related to the total number of cases used in either learning or testing.

As it is seen the best results of testing by using the data not taking part in learning have been obtained for $L = 5$ and we treat this number as an optimal one. Observe that an increase of the rule number to 6 has caused growing the number of misclassifications made on the testing data.

In the second series of experiments we have replaced the classification mode by the more demanding regression and tried to adjust the parameters of the TSK network to match the target concentration values of each gas. This time the best results have been obtained with 12 inference rules. Such a network was adapted to the learning data (250 labeled samples) and then tested on the rest samples (170 labeled samples). The learning and testing procedure have been repeated 50 times with different randomly chosen learning and testing data partitions. To find the objective quality of the neural estimation we have defined two error measures – the general and relative measures of the average absolute discrepancy between the target value and actual estimated value in all trials. The first error is expressed in ppm

$$\Delta = mean(abs(\mathbf{y} - \mathbf{d})). \tag{17}$$

The other is the relative percentage error defined for all trials as

$$\varepsilon = \frac{\text{mean}(\text{abs}(\mathbf{y} - \mathbf{d}))}{\text{mean}(\mathbf{d})} 100\%, \tag{18}$$

where **y** represents the vector of the actual output signals of the neural estimator and **d** – the target vector of 170 measured values.

The graphical illustration of statistical results of testing the system on the data not taking part in learning in all 50 trials of numerical experiments for 4 gases are presented in Fig. 4. The results of testing are varying for different gases. The significant value of variance is mainly due to the small number of available data used in experiments. However, most of trials resulted in relative errors far below 10%.

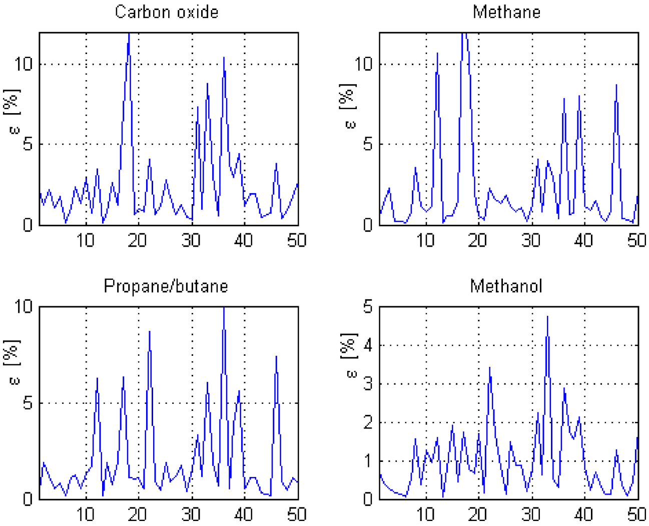

Fig. 4. The relative errors of estimation of concentration of all 4 gases in 50 experiments at random choice of learning and testing data (the testing data results only)

Table 2
The performance of the TSK-based classification in the electronic nose application

Number of inference rules	Number of misclassifications in learning	Percentage of misclassifications in learning	Number of misclassifications in testing	Percentage of misclassifications in testing
3	8	3.64%	12	6.00%
5	5	2.27%	3	1.50%
6	4	1.82%	10	5.00%
8	2	0.91%	7	3.50%
12	0	0.00	6	3.00%

Table 3 presents the statistical results of testing in the form of mean absolute difference Δ between the actual estimation and its respective target (in ppm) and also the relative percentage error ε (in %) corresponding to the results of Fig. 4 (all values calculated for 50 runs of the system on the randomly selected data for learning and testing).

Table 3
The statistical results of testing the TSK-based regression in the electronic nose at estimation of concentration of 4 gases

Gas	Δ [ppm]	$std(\Delta)$ [ppm]	ε [%]	$std(\varepsilon)$ [%]
Carbon oxide	7.23	7.02	2.37%	2.61%
Methane	6.63	7.05	2.19%	2.98%
Propane/butane	5.81	5.88	1.95%	2.13%
Methanol	3.39	3.21	0.99%	0.94%

The mean average errors for all gases obtained in 50 trials of learning and testing the system using different composition of learning and testing data remain on reasonable level (below 2.5%) and the average absolute errors are below 7.5 ppm.

Figure 5 presents the distribution of the estimation errors of the testing data (in ppm) of all gases in one run of experiment and show how these errors are distributed for all 170 testing samples. We can see that only few samples have been estimated with a larger error, while the most of them with small one.

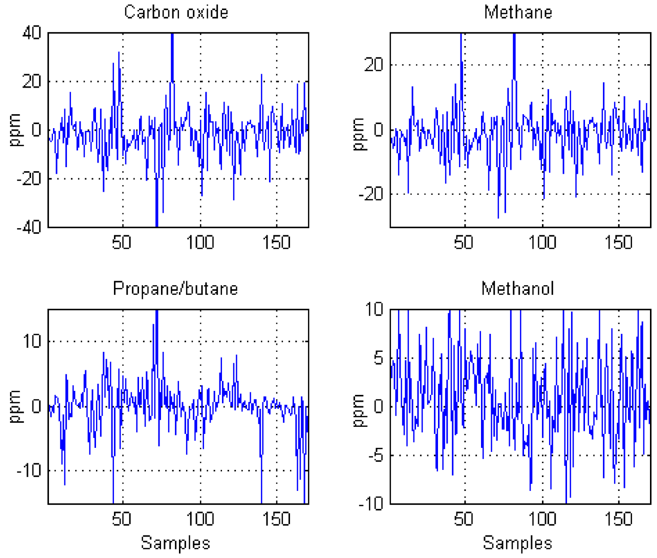

Fig. 5. The distribution of estimation errors for the testing samples in one run of experiment

5. Conclusions

The paper has presented the simplified structure of the Takagi-Sugeno-Kang neuro-fuzzy network. The most important point differing it from the classical ANFIS system is that the inference rules are created on the base of the self-organization principle. In this approach the complexity of the network is controlled by the self-organization mechanism and has no direct connection with the dimension of the input vector **x**. Moreover we avoid the rules which are placed in the regions deprived of data. The use of the Gustafson-Kessel clusterization method allows also to estimate the optimal or suboptimal number of rules and to place their centers in favorable positions. Thanks to this the learning process of the TSK network is simplified and may be done very quickly.

It was shown that the proposed structure of the neuro-fuzzy modified TSK network can be successfully applied in solving the approximation problems of multidimensional data, including regression and classification tasks. We have shown its practical usability in calibration and operation stages of the electronic nose application.

REFERENCES

[1] J.R. Jang, C.T. Sun, and E. Mizutani, *Neuro-fuzzy and Soft Computing*, Prentice Hall, New York, 1997.

[2] L. Rutkowski, *Flexible Neuro-Fuzzy Systems*, Kluwer Academic Publisher, Boston, 2004.

[3] L.X. Wang, *Adaptive Fuzzy Systems and Control: Design and Stability Analysis*, Prentice Hall, New York, 1994.

[4] R. Babuska and H.B. Verbruggen, "Constructing fuzzy models by product space clustering", in *Fuzzy Model Identification*, ed. H. Hellendoorn and D. Driankov, pp. 53–90, Springer, Berlin, 1998.

[5] J. Kacprzyk, *Multistage Fuzzy Control*, WNT, Warszawa, 2001 (in Polish).

[6] R. Yager and D. Filev, *Essentials of Fuzzy Modeling and Control*, Wiley, New York, 1994.

[7] H.J. Zimmermann, *Fuzzy Set Theory and Its Applications*, Kluwer, Boston, 1985.

[8] T. Takag and M. Sugeno, "Fuzzy identification of systems and its application to modeling and control", *IEEE Trans. SMC* 15, 116–132 (1985).

[9] Y. Lin and G. Cunningham III, "A new approach to fuzzy neural system modeling", *IEEE Trans. Fuzzy Systems* 3, 190–198 (1995).

[10] H. Ishibuchi, "Development of fuzzy neural networks", in *Fuzzy Modelling: Paradigms and Practice*, ed. W. Pedrycz, pp. 185–202, Kluwer Academic Publishers, Boston, 1996.

[11] N.R. Pal and J.C. Bezdek, "On cluster validity for the fuzzy c-means model", *IEEE Trans. Fuzzy Systems* 3, 370–379 (1995).

[12] D. Gustafson and W. Kessel, "Fuzzy clustering with a fuzzy covariance matrix", *Proc. IEEE CDC* 1, 761–766 (1976).

[13] A. Gath and B. Geva, "Unsupervised optimal fuzzy clustering", *IEEE Trans. PAMI* 7, 773–781 (1989).

[14] L. Tran Haoi and S. Osowski, "Neuro-fuzzy TSK network for approximation of static and dynamic functions", *Control and Cybernetics* 31, 309–326 (2002).

[15] S. Osowski, L. Tran Hoai, and K. Brudzewski, "Neuro-fuzzy TSK network for calibration of semiconductor sensor array for gas measurements", *IEEE Trans. on Measurements and Instrumentation* 53, 330–337 (2004).

[16] *Matlab User Manual*, Natick, 2012.