Gas Mixture Analysis using Sensor Array and Neuro-Fuzzy Networks

Nima Saffarpour¹  Tomasz Sobanski²

Abstract—This contribution describes a method for discrimination and quantification of different gas mixture using neuro-fuzzy methods. The system includes array of sensors, data acquisition system, computer and neuro-fuzzy network implemented in software. With this method users can detect and classify gases and odors toward achieving electronic nose which has several applications in environmental, medical and food industries.

I. INTRODUCTION

There are many instances where it is necessary or desirable to determine concentrations of individual components in a mixture of gases. This information may be necessary, e.g. to control a chemical process or to monitor the safety of a gas environment in underground mine or to control the quality of materials in food industries. The traditional approach to this sensing problem was to build individual sensors which responds exclusively to each component in the mixture. In such a solution, each sensor’s output respond to concentration of particular a component. Gas sensors (semiconductor oxide) are widely used as convenient means for recognizing and estimating the flammable gas in the air. One of the main problems with this type of gas sensor is their lack of selectivity or only partial selectivity. A certain degree of selectivity can be improved either by operating at different temperatures, or by using catalysts. Another attractive way to achieve this selectivity is to use the array of sensing elements. The task of signal processing systems for a gas sensor system is to classify the recognized gases and to calculate their concentration. Therefore, the signal processing of sensor signals is two folds problem, i.e. to classify the gas and to calculate its concentration. In this work the attention was paid to benzene, cyclohexane, acetone, chloroform, ethanol, acid acetic and ethyl benzene. These components have a great meaning for industry. All of these compounds are used in the production of a wide range of chemical products and may be highly toxic by ingestion, inhalation and skin contact. It was interesting to investigate if gas sensor array could be applied for determination of these compounds in a mixture. For this task different approaches have been proposed. We may classify them into two main categories: parametric and non-parametric approaches [10]. Parametric methods rely upon obtaining or estimating the probability density function of the parameters used to characterize the response of a system. In general these methods require considerable effort in establishing a large database, although different techniques are sometimes used to model smaller data sets, with a reduced confidence level. Conversely, non-parametric methods require no assumption about fundamental statistical distributions of the data. Examples are preprocessing techniques such as scaling, normalizing, correlating and feature weighting. These all involve the transformation of data to promote and display underlying patterns within the data. There are also two types of non-parametric learning or classification method: supervised and unsupervised. Unsupervised methods make no prior assumption about the sample classes but try to separate groups or clusters; these are often termed cluster analysis. In contrast, supervised methods involve the learning of data based on advance knowledge of the classification, e.g. principal component [6],[9] and [2] or artificial neural networks [9], [3], [11], [5], [14] and [15]. They generally involve the systematic determination of optimal rules, algorithms or paradigms in order to classify other unknown samples.

This paper describes the application of a semiconductor sensor array and a neuro-fuzzy network for discrimination and quantification of organic compounds in the mixtures. In the following sections first the elements of the measurement system and its issues are introduced, then theoretical foundation of neuro-fuzzy networks and particularly adaptive network based fuzzy inference system will be introduced and finally the achieved results will be shown and compared with previous work.

II. THE COMPONENTS OF SYSTEMS

Because the mixture of organic compounds often includes components which are very similar to each other
with respect to their chemical constitution, they are analyzed with the use of a sensor array. It is caused by the fact that a single sensor has limited selectively and its answer to similar compounds does not differ very much. The use of sensor arrays is an idea to analyze not a single answer of a non-selective sensor but pieces of information from several sensors (differentiated with references to gas sensor layer). By this way we have much more detailed information because it is based on the difference in sensors answer. Because of the high number of phenomena happening at a sensor surface on its gas sensitive parameters, it is difficult to develop a parametric algorithm for answer analysis [4]. The situation is further complicated by the fact that during the analysis data are used come from several sensors not just only one, long term changes of parameters of the layer are the next difficulties. All the above mentioned difficulties cause that a precise description of the contents of mixture with the use of parametric methods is unrealisable. It is mainly due to the fact that no complex model has been developed so far which would describe all phenomena occurring both at the surface and the inside the gas sensitive layer [3].

The experimental system is composed by a sensor array, a test chamber, a calibrating unit, a data acquisition card, and a personal computer. The sensor array was made up of six commercial semiconductor gas sensors-TGS (manufactured by Figaro electronics, Japan). All sensors were placed in a glass test chamber.

Fig. 2 and 3 show the response of array of sensors versus the concentration of single component (in this case benzene and ethylbenzene). As it can be seen a linear model may be applied at low level concentrations of benzene or ethylbenzene. This assumption of linearity not only considerably simplifies the characterization of gas sensor but also permits the implementation of linear pattern recognition models, for example, the principal of superposition. However, at higher component concentrations the response of tin dioxide, and more generally, the metal oxide gas sensors are non-linear. The situation is much more difficult if we have multi component systems i.e. mixture of gases at different concentrations and also different level of humidity and temperature. This is mainly because of cross sensitivities exist in current generation solid state gas sensors. Figure 4 shows the responses of sensors against mixture of benzene and ethylbenzene.

III. THE PRINCIPLE OF NEURO-FUZZY NETWORK

The origin of neural fuzzy inference systems is to incorporate neural concepts, such as learning and parallelism, into fuzzy logic inference systems. Neural fuzzy inference systems realize fuzzy inference. The architecture of the systems are parallel, and they exploit the same learning algorithms, which are used with neural networks. As already mentioned, the effectiveness of the fuzzy models representing nonlinear input-output relationships depends on the fuzzy partition of the input-
output space. Therefore, the tuning of membership functions becomes an important issue in fuzzy modeling. Since this tuning task can be viewed as an optimization problem, neural network and genetic algorithms offer a possibility to solve this problem. A straightforward approach is to assume a certain shape for the membership functions which depends on different parameters that can be learned by a neural network[1]. We require a set of training data in the form of correct input-output pairs and a specification of the rules including a preliminary definition of the corresponding membership functions. Let us consider multi-input, one-output system to be modeled as the fuzzy system. Donate the input vector by $X = [x_1, x_2, \ldots, x_N]^T$ and the output signal by $y$. In neuro-fuzzy network the most often used model is TSK model which has been introduced in previous chapter. The general form of TSK inference rule can be stated as following[8]:

If $x_1$ is $A_1$ and $x_2$ is $A_2$ and ... and $x_n$ is $A_n$ then $y = f(x_1, x_2, \ldots, x_N)$ which can be written in the vector form as follows:

If $X$ is $A$ then $y = f(x)$ where $f(X) = f(x_1, x_2, \ldots, x_N)$ is crisp function. The premise if $x_i$ is $A_i$ is implemented by fuzzifier, for the case of gaussian membership function:

$$
\mu_a(x_j) = e^{-\frac{1}{2} \left(\frac{x_j - c_i}{\sigma_i}\right)^2}
$$

(1)

The antecedent part of first-order TSK can be written as:

$$
y = f(x) = p_0 + \sum_{i=1}^{N} p_i x_i
$$

(2)

where $p_0, p_1, \ldots, p_N$ are crisp numbers adjusted in the learning process.

The second order neuro-fuzzy system, known as the Wang-Mendel model, is the system implementing the zero order TSK function of the form

$$
y = f(X) = p_0
$$

(3)

In this model $p_0$ may be interpreted as the center of the consequent part of the fuzzy inference rule. Applying the aggregation of vector $X$ in the form of algebraic multiplication

$$
w_k = \mu_A^{(k)} (X) = \prod_{j=1}^{N} \left[ e^{-\frac{1}{2} \left(\frac{x_j - c_i}{\sigma_i}\right)^2} \right]
$$

(4)

and considering $M$ rules of inference we get the resulting system description in the form of weighted mean of all rules:

$$
y(X) = \frac{\sum_{k=1}^{M} w_k f_k(X)}{\sum_{k=1}^{M} w_k}
$$

(5)

where $f_k(X)$ is either of the form equation 2 or equation 3 and the normalization condition holds:

$$
\sum_{i=k}^{M} w_i = 1
$$

(6)

The adjusted parameters of the system are nonlinear parameters $(c_i, \sigma_i)$ and linear parameters (weights $p_i$) of TSK functions.

IV. ADAPTIVE NETWORK-BASED FUZZY INFERENCE SYSTEM (ANFIS)

The adaptive network based structure corresponding to the equation 7 is shown in figure 5. This is a 5 layer structure[7], in which:
- layer1: performs fuzzification $\mu^{(k)}_{A_i}(x_i)$ (parametric layer of parameters $c_i^{(k)}$ and $\sigma_i^{(k)}$)
- layer2: aggregation $w_k = \mu^{(k)}_{A_k}(X)$ (nonparametric layer)
- layer3: is the generator of TSK functions $f_k(X) = p_{k0} + \sum_{j=1}^{N} p_{kj} x_j$ (parametric layer of parameters $p_{kj}$ for $k = 1, 2, ..., M$ and $j = 0, 1, 2, ..., N$)
- layer4: summation layer (nonparametric layer) performing operation $y(X) = f(X) = \frac{1}{E}$.

In this way the neuro-fuzzy structure implements the following function

$$y(X) = \frac{\sum_{k=1}^{M} \prod_{j=1}^{N} \mu^{(k)}_{A_k}(x_j)[p_{k0} + \sum_{j=1}^{N} p_{kj} x_j]}{\sum_{r=1}^{M} \prod_{j=1}^{N} \mu^{(r)}_{A_r}(x_j)}$$ (7)

For $N$ input variables and $M$ rules there are $M(N+1)$ adjustable parameters of the ANFIS functions for each output. On the other hand each $\mu(x) \text{ corresponds to } c \text{ and } \sigma \text{ parameters of the nonlinear membership function, described by equation 1.}$

Fig. 5. The structure of ANFIS network

The hybrid learning algorithm belongs to the the class of supervised learning, that performs the minimization of the cost function defined by using the Euclidean measure[13]:

$$E = \frac{1}{p} \sum_{i=1}^{p} (y(X^{(i)}) - d^{(i)})^2$$

where $p$ is the number of learning pairs $(X, d)$ and $d$ is the desired output. The adapted parameters are divided into 2 groups: the linear parameters $p_{kj}$ of the TSK functions and nonlinear parameters of fuzzifiers $c_{x_i}$ and $\sigma_i$.

The hybrid algorithm is composed of two separate stages:
- Stage 1: adaptation of linear parameters $p_{kj}$ of the TSK functions at fixed values of the nonlinear parameters.
- Stage 2: the adaptation of nonlinear parameters at the fixed values of linear parameters.

These steps are repeated many times until stabilization of all parameters of the network. In stage 1 we exploit the observation, that $y(X)$ is the linear function of $p_{kj}$

$$y(X) = \sum_{k=1}^{M} w_k^{'} p_{k0} + \sum_{j=1}^{N} p_{kj} x_j$$ (8)

where

$$w_k^{'} = \frac{1}{\sum_{r=1}^{M} \prod_{j=1}^{N} \mu^{(r)}_{A_r}(x_j)} \sum_{k=1}^{M} \prod_{j=1}^{N} \mu^{(k)}_{A_k}(x_j)$$ (9)

Since the weights $w_k^{'}$ at fixed values of non-linear parameters of the membership functions are constant for $k = 1, 2, ..., M$, at $p$ learning pairs $(X^{(i)}, d^{(i)}) (i = 1, 2, ..., p)$ we got the system of $p$ linear equations, which can be written as follows:

$$AP = d$$ (10)

where

$$A = \begin{bmatrix}
    w_{11}^{'} & w_{21}^{'} x_1^{(1)} & \cdots & w_{11}^{'} x_N^{(1)} \\
    w_{21}^{'} & w_{22}^{'} x_1^{(2)} & \cdots & w_{21}^{'} x_N^{(2)} \\
    \vdots & \vdots & \ddots & \vdots \\
    w_{p1}^{'} & w_{p2}^{'} x_1^{(p)} & \cdots & w_{p1}^{'} x_N^{(p)} \\
\end{bmatrix}$$ (11)

$$P = \begin{bmatrix}
p_{10} \\
p_{1N} \\
\vdots \\
p_{MN}
\end{bmatrix}$$ (12)

$$d = \begin{bmatrix}
d^{(1)} \\
d^{(2)} \\
\vdots \\
d^{(p)}
\end{bmatrix}$$ (13)
In these equations the coefficient \( w_{ki} \) represent the firing strength of i-th rule at the presentation of k-th input vector \( X \) to the network.

The matrix form of the last equation has simple solution, that can be determined using pseudo inverse of matrix \( A \), denote here by \( A^+ \):

\[
P = A^+d
\]

(14)

In the second stage, after fixing linear parameters \( p_{ij} \) the output signal \( y^{(i)} \) are calculated for \( i = 1, 2, \ldots, p \)

\[
y = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(p)} \end{bmatrix}
\]

(15)

\[y = AP \]

(16)

and then the error \( e = y - d \).

The adaptation of nonlinear parameters can be done using steepest descent method and back propagation of the error, i.e.

\[c_j^{(k)}(n+1) = c_j^{(k)}(n) - \eta \frac{\partial E(n)}{\partial c_j^{(k)}} \]

(17)

\[\sigma_j^{(k)}(n+1) = \sigma_j^{(k)}(n) - \eta \frac{\partial E(n)}{\partial \sigma_j^{(k)}} \]

(18)

where \( n \) is the iteration number, the generation of the gradient component is relatively simple, since the error function is given in an explicit form. These relations can be presented in the following form:

\[\frac{\partial E}{\partial c_j^{(k)}} = (y(X) - d)[p_{r0} + \sum_{j=1}^{N} p_{rj} x_j] \frac{\partial w_{i}'}{\partial c_j^{(k)}} \]

(19)

\[\frac{\partial E}{\partial \sigma_j^{(k)}} = (y(X) - d)[p_{r0} + \sum_{j=1}^{N} p_{rj} x_j] \frac{\partial w_{i}'}{\partial \sigma_j^{(k)}} \]

(20)

The derivatives \( \frac{\partial w_{i}'}{\partial c_j^{(k)}} \) and \( \frac{\partial w_{i}'}{\partial \sigma_j^{(k)}} \) appearing in these relations are determined on the basis of equation 8 and equation 9 and take the form:

\[\frac{\partial w_{i}'}{\partial c_j^{(k)}} = \sum_{j'=1}^{N} \frac{m(x_j) l(x_{j'})}{|m(x_j)|^2} \delta_{j'j} \frac{\partial \mu^{(k)}_{A_i}(x_j)}{\partial c_j^{(k)}} \]

(21)

\[\frac{\partial w_{i}'}{\partial \sigma_j^{(k)}} = \sum_{j'=1}^{N} \frac{m(x_j) l(x_{j'})}{|m(x_j)|^2} \delta_{j'j} \frac{\partial \mu^{(k)}_{A_i}(x_j)}{\partial \sigma_j^{(k)}} \]

(22)

for \( r = 1, 2, \ldots, M \), where \( \delta_{jk} \) means kro- necker delta and \( l(x_{j'}) = \prod_{j=1}^{N} \mu^{(k)}_{j_{\text{max}}}(x_{j'}) \cdot \mu^{(k)}_{j_{\text{min}}}(x_{j'}) \) \( m(x_j) = \sum_{k=1}^{M} \prod_{j=1}^{N} \mu^{(k)}_{j_{\text{max}}}(x_{j}) \).

The above equations express, in analytical form, the quantities required at the adaptation process of the neuro-fuzzy TSK network.

V. SIMULATION RESULTS

The neuro-fuzzy structure has been applied for qualification and quantification of gas mixture component by applying he array of sensors. In a conventional fuzzy inference system, the number of rules is determined by an expert who is familiar with the target system to be modeled. In our simulation, however, the type and the number of membership functions assigned to each input variable is chosen empirically, that is by plotting the data sets and examining them visually or simply by trial and error.

In our simulation we have six inputs, which receive the responses of the sensor array. We have examined 4 different mixtures, i.e. benzene + ethyl-alcohol, acetone + chloroform, benzene + cyclohexane and ethanol + acid acetic with different level of humidity(0, 4, 10, 100%).

In all cases the 50% of data is used as training data set, 25% as checking (or validation) data set and 25% as testing data set. The checking or validation data is used to prevent overfitting of the training data set. Overfitting can be detected when the checking error starts increasing while the training error is still decreasing. The ANFIS structure used here contains 4 rules in form of:

- Rule1 = If \( x_1 \) is \( A_{11} \) and \( x_2 \) is \( A_{12} \) and \( x_3 \) is \( A_{13} \) and \( x_4 \) is \( A_{14} \) and \( x_5 \) is \( A_{15} \) and \( x_6 \) is \( A_{16} \) then \( y_1 = p_{10} + p_{11} x_1 + p_{12} x_2 + p_{13} x_3 + p_{14} x_4 + p_{15} x_5 + p_{16} x_6 \)
- Rule2 = If \( x_1 \) is \( A_{21} \) and \( x_2 \) is \( A_{22} \) and \( x_3 \) is \( A_{23} \) and \( x_4 \) is \( A_{24} \) and \( x_5 \) is \( A_{25} \) and \( x_6 \) is \( A_{26} \) then \( y_1 = p_{20} + p_{21} x_1 + p_{22} x_2 + p_{23} x_3 + p_{24} x_4 + p_{25} x_5 + p_{26} x_6 \)
- Rule3 = If \( x_1 \) is \( A_{31} \) and \( x_2 \) is \( A_{32} \) and \( x_3 \) is \( A_{33} \) and \( x_4 \) is \( A_{34} \) and \( x_5 \) is \( A_{35} \) and \( x_6 \) is \( A_{36} \) then \( y_1 = p_{30} + p_{31} x_1 + p_{32} x_2 + p_{33} x_3 + p_{34} x_4 + p_{35} x_5 + p_{36} x_6 \)
- Rule4 = If \( x_1 \) is \( A_{41} \) and \( x_2 \) is \( A_{42} \) and \( x_3 \) is \( A_{43} \) and \( x_4 \) is \( A_{44} \) and \( x_5 \) is \( A_{45} \) and \( x_6 \) is \( A_{46} \) then \( y_1 = p_{40} + p_{41} x_1 + p_{42} x_2 + p_{43} x_3 + p_{44} x_4 + p_{45} x_5 + p_{46} x_6 \)
- Rule5 = If \( x_1 \) is \( A_{51} \) and \( x_2 \) is \( A_{52} \) and \( x_3 \) is \( A_{53} \) and \( x_4 \) is \( A_{54} \) and \( x_5 \) is \( A_{55} \) and \( x_6 \) is \( A_{56} \) then \( y_1 = p_{50} + p_{51} x_1 + p_{52} x_2 + p_{53} x_3 + p_{54} x_4 + p_{55} x_5 + p_{56} x_6 \)
- Rule6 = If \( x_1 \) is \( A_{61} \) and \( x_2 \) is \( A_{62} \) and \( x_3 \) is \( A_{63} \) and \( x_4 \) is \( A_{64} \) and \( x_5 \) is \( A_{65} \) and \( x_6 \) is \( A_{66} \) then \( y_1 = p_{60} + p_{61} x_1 + p_{62} x_2 + p_{63} x_3 + p_{64} x_4 + p_{65} x_5 + p_{66} x_6 \)

and 4 gaussian membership functions assigned to each input. Here are some important statistics about the neuro-fuzzy model:

- The total number of input: 6
- The number of fuzzy rules: 4
- The number of membership functions : 4
- The total number of fitting parameters : 76
The number of premise (non linear) parameters: 48

The number of consequent (linear) parameters: 28

There are 48 non linear parameters, because there are 6 inputs and for each input there are 4 membership functions, which means we have totally 24 membership functions, and each membership function has 2 parameters, \( c_i, \sigma_i \), therefore there are 48 non linear adaptation parameters. In the consequent part of each rule there are 7 linear parameters, \( D_i, B_i^1, B_i^2, B_i^3, B_i^4, B_i^5, B_i^6 \) which should be tuned during the training process. Therefore, for a system with two outputs, there are 152 adaptation parameters. If we compare this with the adaptation parameters of best neural network structure in our simulation [12], which has produced best results, i.e. 6-12-12-2 structure, we observe it has 266 adaptation parameters (including 240 connection weights and 26 biases). This increases learning epoch and slows down the training time.

Table 1 shows the maximum and average percentage of error based on the range of input data (see Eq. 23 and 24), which also has been compared with the result of neural network [12]. Comparing the results shows that, in almost all cases neuro-fuzzy network shows lower recognition error. In Fig. 10 to 13 the level of error in ppm have been shown in which the error level are, except one case, always lower than 0.15 ppm. We have sketched both error levels of neural networks in the same diagram in order to compare the output errors.

Average percentage error:

\[
\text{Average percentage error} = \frac{1}{P} \sum_{i=1}^{P} \frac{|T(i) - O(i)|}{|T(i)|} \times 100\% 
\]

Maximum percentage error:

\[
\text{Maximum percentage error} = \text{Max} \left( \frac{|T(i) - O(i)|}{|T(i)|} \times 100\% \right) 
\]

As another analysis of results, we apply linear regression analysis of outputs which can be seen in the Fig. 6 to 9 show the regression analysis of neuro-fuzzy network outputs. The parameter \( R \) shows correlation coefficient between output of network and real value. It is a measure of how well the variation in the output is explained by desired values. If this number is equal to 1, then there is a perfect correlation between outputs and the real values. In our case this value is always more than 0.98 which shows very good correlation.

Finally, we have compared the changes of membership functions on each input before and after training, it means we would like to see the impact of training on each membership function.

<table>
<thead>
<tr>
<th>Mixture</th>
<th>Neural Network Max</th>
<th>Mean</th>
<th>Neuro-Fuzzy Max</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>benzene</td>
<td>10.906</td>
<td>0.1523</td>
<td>7.333</td>
<td>0.8176e-7</td>
</tr>
<tr>
<td>ethylbenzene</td>
<td>12.894</td>
<td>0.1740</td>
<td>13.405</td>
<td>9.631e-8</td>
</tr>
<tr>
<td>acetone</td>
<td>9.122</td>
<td>0.039</td>
<td>8.215</td>
<td>8.600e-6</td>
</tr>
<tr>
<td>chloroform</td>
<td>11.00</td>
<td>0.235</td>
<td>8.117</td>
<td>1.300e-6</td>
</tr>
<tr>
<td>benzene</td>
<td>12.49</td>
<td>0.036</td>
<td>9.18</td>
<td>3.250e-6</td>
</tr>
<tr>
<td>cyclohexane</td>
<td>15.264</td>
<td>0.352</td>
<td>11.837</td>
<td>4.010e-6</td>
</tr>
<tr>
<td>ethanol</td>
<td>23.38</td>
<td>0.173</td>
<td>18.95</td>
<td>2.270e-6</td>
</tr>
<tr>
<td>acetic acid</td>
<td>19.041</td>
<td>0.71</td>
<td>15.91</td>
<td>2.043e-6</td>
</tr>
</tbody>
</table>

Table 1

The comparison of maximum and average output error between neural network and neuro-fuzzy network
VI. CONCLUSION

The paper has discussed the modeling of multi dimensional data using neuro-fuzzy structure of ANFIS type. It has been shown in this paper that with appropriate data preprocessing and efficient learning procedure the neuro-fuzzy networks are capable of making the recognition of the gas components and estimation of their concentration with good accuracy, and may be implemented in low cost universal measurement systems. The neuro fuzzy paradigm which has been proposed in this paper for gas analysis shows better performance than other methods in terms of recognition error. The system is flexible, adaptive and intuitive plausible and can model non-linear functions of arbitrary complexity.
Fig. 13. Level of output error for chloroform

System fuzzy classifier: 6 inputs, 1 outputs, 4 rules

Fig. 14. The scheme of neuro-fuzzy classifier for benzene before training

REFERENCES


