14th IMEKO TC10 Workshop Technical Diagnostics
New Perspectives in Measurements, Tools and Techniques
for system's reliability, maintainability and safety
Milan, Italy, June 27-28, 2016

Ambiguity groups detection in analog systems diagnostics using Self-Organizing Maps

Piotr Bilski[1]

[1] *Institute of Radioelectronics and Multimedia Technologies, Warsaw University of Technology, ul. Nowowiejska 15/19, 00-650 Warsaw, Poland, pbilski@ire.pw.edu.pl*

***Abstract* – The paper presents the application of Self-Organizing Maps (SOM) to the ambiguity groups detection in the analog system. This type of neural network is able to find dependencies in data, indicating groups of similar examples in the data set used for training the classifier or the regression machine. Various configurations of the network were implemented and compared. The ability to detect ambiguity groups was verified on the model of the induction machine. Results show the efficiency of the approach, able to identify examples difficult to distinguish by the fault detection and location scheme.**

I. INTRODUCTION

The artificial intelligence (AI) has been significantly developed for the diagnostics of analog systems. Multiple approaches, belonging to Simulation Before Test (SBT) and Simulation After Test (SAT) were successfully implemented, proving significant advantages of such algorithms: automated data processing, ability to work in the uncertainty conditions, extraction of knowledge in large data sets and generalization, i.e. the correct reaction to phenomena not present during the training. One of constantly existing problems is the appropriate preparation of training data for the AI method to enable all these advantages. The examples (labeled vectors of symptoms measured at the accessible and partially accessible nodes) should be carefully selected to represent as many states of the System Under Test (SUT) as possible. This is a difficult task, requiring the deep understanding of the SUT work regime. The obtained data set should be evaluated regarding the ability to distinguish different states of the analyzed object. This process may reveal examples similar regarding the selected criterion, but belonging to different categories.

The ambiguity groups (AG) detection [1] is one of the key elements in the diagnostics of analog systems. It determines the ability to distinguish between the particular faults and assess the difficulty of testability in the selected SUT based on the particular set of symptoms. This way it is easier to predict the diagnostic accuracy of the applied algorithm and to verify the usefulness of the symptoms measured in the selected domain. Although mainly considered in the analysis of circuits with low testability [2], it was also applied during monitoring of aerospace systems [3] and during making business decisions [4]. This justifies extending its applications to other systems.

The paper presents the application of the Self-Organizing Map to the AG detection in the analog systems. From the AI perspective the process is seen as grouping similar SUT states, which are similar regarding the selected measure. The SOM [5] is well established clustering method, therefore can be used for this purpose. The key problem in its application is the selection of the network configuration, leading to the generation of the optimal groups. Their quality should be further confirmed by the classification or regression method used to distinguish between the faults.

The structure of the paper is as follows. In section II the AG are defined and the possibility of detecting them using the AI-based approaches introduced. Section III presents the application of the SOM to the introduced task. In Section IV the selected SUT, i.e. the induction motor simulated for the diagnostics is presented. Section V contains results of the experiments. In Section VI conclusions and prospects of the selected methodology are discussed.

II. AMBIGUITY GROUPS IN DIAGNOSTICS

Ambiguity groups are phenomenon inherent to the diagnostics of analog systems. The idea of detecting and identifying faults is based on the non-intrusive monitoring of available response signals $\boldsymbol{y}(\boldsymbol{p},t)$ collected at accessible nodes after introducing excitations $\boldsymbol{x}(t)$ on its inputs (Fig. 1). The form of responses depends on the values of SUT parameters $\boldsymbol{p}=\{p_1,\dots,p_k\}$. The information obtained from $\boldsymbol{y}(\boldsymbol{p},t)$ allows for detecting the state of the SUT. In the presented research parametric faults [6] are considered, where the structure of the faulty SUT is identical to the correctly working one, but changes in parameters' values make its operation not compliant to the designer's expectations. Usually, characteristics of the system have the form of the vector of symptoms $\boldsymbol{s}=\{s_1, \dots, s_m\}$ extracted from measured responses. The proper selection of symptoms is the key to the accurate fault detection, localization and identification. The combination of $\boldsymbol{s}$ and the information about the actual state c enables finding relations between these two during the knowledge extraction and verification of the selected method efficiency. Simulation of the SUT in the computing environment enables generation of multiple labeled examples $\boldsymbol{e}$, i.e. vectors of symptoms supplemented by the actual state of the SUT: $\boldsymbol{e}_i=\{\boldsymbol{s}_i,c_i\}$.

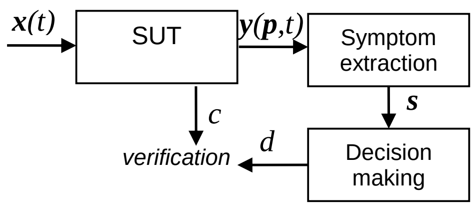

Fig. 1. Data-driven diagnostic decision making.

The state c_i considered here is the discrete identifier of the particular configuration of parameters $\boldsymbol{p}$. Such approach defines the diagnostics as the classification task, requiring the correct identification of the integer number, representing the SUT state. To simplify the procedure, single faults are considered, where only one parameter is beyond the tolerance margin at a time. The assignment of the particular example to the state depends on the discretization strategy (i.e. dividing the range of changes in the particular parameter). For instance, if the tolerance margins for the parameter p_1 are 10% of its nominal value, three categories may be representing its state (Fig. 2): for the value within tolerances, greater than and lesser than the nominal range. The additional partitioning of data is possible, leading to the greater number of categories. To describe the SUT states, the fault codes were used, consisting of the number of the analyzed parameter and the intensity (both positive and negative) of the fault. For the example from Fig. 2, the generated codes will be as follows: "-11", "0", "11". The "0" code is the nominal state, as all other parameters are also within the tolerance margin [4].

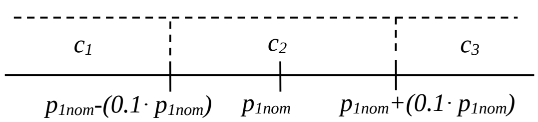

Fig. 2. Assignment of categories to the SUT states.

The n examples form data sets $\boldsymbol{L}$, used by the AI-based algorithm to extract knowledge about the behavior of the system based on it symptoms' values.

The AG analysis was considered in the past, mainly as the theoretical [7] or numerical [2] problem. The applied approaches include the Singular Value Decomposition (which is the standard method for the transformation of data) or the probabilistic reasoning [8]. The AI, although widely used in the contemporary diagnostics, is the new concept [9]. This justifies applications of the selected algorithms for the task. The data-driven approaches (such artificial neural networks – ANN [11], or fuzzy logic [10]) are considered as the most useful and versatile in detecting and locating faults. However, their accuracy depends on the quality of data provided during the training, namely:

- The number of training examples in $\boldsymbol{L}$ and their coverage of the SUT states, including the nominal one (where all parameters $\boldsymbol{p}$ are within tolerance margins) and the most probable faults.
- The set of symptoms $\boldsymbol{s}$ selected for the analysis.
- The similarity between the particular states regarding the selected symptoms.

Optimizing the indicated aspects influences the testability of the system [12], i.e. the ability to identify the SUT conditions. It is desired to maximize the number of correctly determined states. Unfortunately, the existence of AG makes it difficult. The latter are defined as sets of SUT states, indistinguishable from each other:

$$\{c_i,c_j\}:d(\boldsymbol{s}_i)=d(\boldsymbol{s}_j)\wedge c_i\neq c_j \tag{1}$$

The AG contains at least two states. All its members are form pairs with other states exiting in AG, fulfilling the condition (1). Their negative effect on the diagnostic accuracy can be minimized by increasing or modifying the set $\boldsymbol{s}$. This requires changing or adding the analysis domain (time, frequency, or mixed). Another method is to introduce new accessible nodes (if physically possible), at which response signals are recorded. Detecting the AG also allows for evaluating the difficulty of the SUT prior to its analysis by the selected diagnostic method.

The following types of AG may exist in the SUT:

- The group containing the nominal state with any number of faulty ones. This means some faults may be difficult to detect and the SUT could be considered as working properly, posing problem even for the fault detection. This is typically caused by the low sensitivity of some of parameters $\boldsymbol{p}$.
- The group containing various faulty states. This way the fault detection is possible, but its precise location and identification - not.

Because the ability to distinguish between different faults is based only on recorded symptoms, it is possible to miss the fault if the SUT behaves as the nominal one (even if its parameters are beyond tolerances). When the system meets all specifications (regarding the power consumption, values of generated signals, operational temperatures, etc.), this is not considered a problem.

The existence of similar, hardly distinguishable states causes a problem for the AI-based diagnostic module. During training, the algorithm may try to distinguish between different states manifested by similar vectors of symptoms. This leads to the overfitting to data, i.e. focusing on unimportant details, which degrade the diagnostic accuracy. Therefore such states should be detected and indicated for further analysis. The solution is the unsupervised learning scheme, grouping similar examples disregarding their actual categories.

The AI-based AG detection consists in processing the training set $\boldsymbol{L}$ and identifying all examples which might be difficult to distinguish by the intelligent algorithm. The categories c are ignored for the process, allowing to find the actual dependencies in data. Every SUT state is represented by the number of examples, related to various intensity of the fault. Also, existence of the additive noise or measurement uncertainty justify generating multiple instances of symptoms for the same state (when small, random changes of symptoms' values occur for the same set of SUT parameters). This way the AG detection is the identification of similar examples. It is possible that only part of examples belonging to the same state is indistinguishable from the ones belonging to another state (see Fig. 3). This suggests the categories should be divided into subcategories, two easily distinguishable from each other and the third one, forming the actual AG.

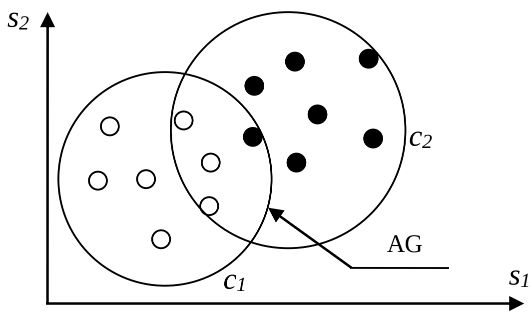

Fig. 3. Illustration of examples from categories c_1 and c_2 belonging to one ambiguity group.

III. SELF-ORGANIZING MAPS

The SOM (Kohonen network) is the grouping algorithm with multiple applications in technical [13] and environmental sciences. This is the one-layered ANN with computing units (neurons) located in the two-dimensional grid (Fig. 4). Each neuron is connected to every input (here the symptom s_i from the vector $\boldsymbol{e}_i$). The AG detection implemented here consists in the repeated presenting all examples from $\boldsymbol{L}$ and modifying the weights of neurons reacting the strongest to the particular example. This way after a number of epochs (presentations of the whole dataset), the selected neurons begin to react on the particular examples, becoming their representatives. All examples triggering the same neuron are one group, with the weights of the neuron as the centroid. If the group components belong to different categories, the AG is located. The training process is evaluated by the quantization error (2), being the mean value of the Euclidean distance between the particular examples and neurons the most adjusted for them (represented by the optimal weight vector $\boldsymbol{w}_i$*). The stopping criterion for the training process is reaching the maximum number of epochs (here 100) or obtaining the minimal acceptable value of e_q.

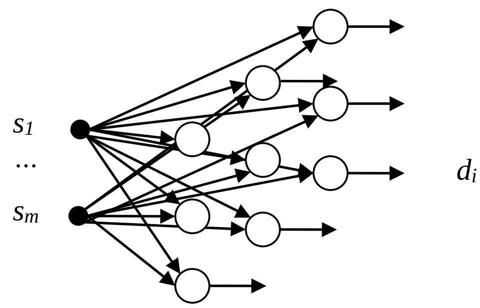

Fig. 4. Structure of the SOM.

$$e_q = \frac{1}{n} \cdot \sum_{i=1}^{n} \left\| \boldsymbol{s}_i - \boldsymbol{w}_i^* \right\|^2 \qquad (2)$$

The network is characterized by multiple parameters, determining its behavior, allowing for obtaining the particular clustering results. The most important for the presented problem, include [14]:

- The number of neurons and their positions on the two-dimensional plain. The SOM neurons may form the square or rectangular shapes. Alternatively, the linear, one-dimensional position of neurons is possible. The location of units is important when the WTM strategy is used (see below).
- The training strategy and the method of determining, which neurons should be trained. The Winner Takes All (WTA) approach allows for modifying weights only of the single winning neuron (here the location of neurons is irrelevant, only their number matters). In the WTM (Winner Takes Most) approach the winning neuron and some of its neighbors are trained. The neighborhood is usually defined as the rectangular (all neurons are trained in the same fashion) or Gaussian (the neurons farther from the winner are trained less).
- Introduction of the conscience mechanism, i.e. switching off the winning neuron for the number of examples' presentations. It was expected to introduce the repeatable results to the training process (which was confirmed during the experiments).
- Learning coefficient η, which determines the speed of training, i.e. the intensity of changing weights of neurons. Usually the compromise must be found between the learning time (shorter for greater values) and the accuracy of the configured network (lower values of e_q are obtained for small values).

All the parameters were evaluated during the SOM implementation, as they influence the number and cardinality of the generated AG. The clustering algorithm was implemented in the Matlab environment, Although there are existing versions of the Kohonen network, in the presented research the author version was used. This allowed for the more detailed parameterization of the method to check its characteristics.

IV. ANALYZED SYSTEM

Asynchronous motors are alternate current machines. Their work regime consists in rotating the rotor inside the changing magnetic field created by the current flowing through the stator. Modeling of such a device is well established [15]. The typical model is described by the equations (2) [16]:

$$\begin{aligned}
\frac{di_{sd}}{dt} &= \frac{\beta}{T_R}\cdot\varphi_{rd} + \beta\cdot n_p\cdot\omega_r\cdot\varphi_{rq} - \gamma\cdot i_{sd} + \frac{1}{\sigma\cdot L_s}\cdot u_{sd} \\
\frac{di_{sq}}{dt} &= \frac{\beta}{T_R}\cdot\varphi_{rq} - \beta\cdot n_p\cdot\omega_r\cdot\varphi_{rd} - \gamma\cdot i_{sq} + \frac{1}{\sigma\cdot L_s}\cdot u_{sq} \\
\frac{d\varphi_{rd}}{dt} &= -\frac{1}{T_R}\cdot\varphi_{rd} - n_p\cdot\omega_r\cdot\varphi_{rq} + \frac{M}{T_R}\cdot i_{sd} \\
\frac{d\varphi_{rq}}{dt} &= -\frac{1}{T_R}\cdot\varphi_{rq} - n_p\cdot\omega_r\cdot\varphi_{rd} + \frac{M}{T_R}\cdot i_{sq} \\
\frac{d\omega}{dt} &= \frac{M\cdot n_p}{J\cdot L_r}\cdot\left(i_{sq}\cdot\varphi_{rd} - i_{sd}\cdot\varphi_{rq}\right) - \frac{C_e}{J}
\end{aligned} \qquad (3)$$

with the electrical parameters: i_{sd}, i_{sq} (*d*- and *q*-axis components of the stator current), φ_{rd}, φ_{rq} (*d*- and *q*-axis components of the rotor flux linkages), T_R (the rotor time constant), n_p (the number of magnetic pole pairs), L_s, L_r, M (stator, rotor and mutual inductances), u_{sd}, u_{sq} (*d*- and *q*-axis components of the stator voltage). The mechanical parameters are: ω_r (the rotor angular speed), σ (the total leakage factor), C_e (torque) and J (inertia). Coefficients β and γ are defined as (7):

$$\beta = \frac{M}{\sigma\cdot L_s\cdot L_r} \qquad \gamma = \frac{R_s}{\sigma\cdot L_s} + \frac{M^2\cdot R_r}{\sigma\cdot L_s\cdot L_r} \qquad (4)$$

The model was simulated in the SIMULINK environment. Seven analyzed parameters of the considered motor and their nominal values were $R_s=2.25\ \Omega$, $L_s=0.1232\ H$, $L_r=0.1122\ H$, $M=0.1118\ H$, $T_R=0.16\ s$, $\sigma=0.09$, $J=0.0504$. The remaining parameters remained unchanged at their nominal values throughout the test: $R_r=0.7\Omega$, $n_p=3$. Each analyzed parameter was assigned seven values, leading to 49 total rows in the data set. The number of categories was, depending on the used discretization scheme (Fig. 2), 15, or 27 (respectively, with two, or four fault categories for every parameter)

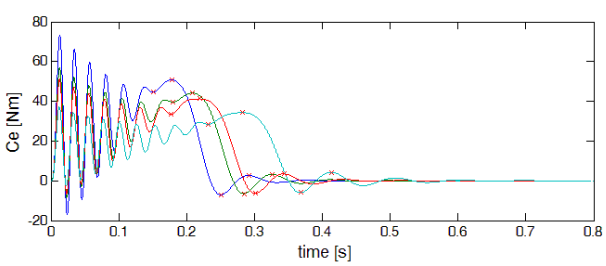

Fig. 5. Induction motor's torque for various values of R_s.

Motor responses were analyzed in the startup phase. Four signals were observed: the stator current vector I_s, torque C_e, angular speed ω, and rotor flux φ_r. From each signal, characteristic features were collected. From the torque (Fig. 5) time instants and values of four extremes before reaching the stable state were extracted. The model of the system was simulated in the Matlab computing environment.

V. EXPERIMENTAL RESULTS

The training set ***L*** of the induction motor was processed by the SOM to determine which examples are difficult to distinguish. All parameters introduced in Section III were verified during the training. The influence of the network topology on the obtained result is in Table 1 (the experiment was conducted for examples labeled with 27 fault codes). It contains the rectangular SOM topology, the minimum obtained value of e_q, the number of epochs required to obtain the latter, the number of detected AG and the range for training neurons if the WTM strategy is applied (1 means only direct neighbors of the winner are trained, decreasing the range of weights' changes).

In all presented experiments the aim of the SOM training was minimization of the quantization error. During the process, the e_q value starts at the relatively high level and within 10 to 50 epochs (depending on the parameters of the network) decreases to the acceptably low value. Unfortunately, for some configurations the training process does not converge, getting stuck at the higher e_q value, or demonstrates the stochastic behavior. In such a case, the AG may significantly differ from the ones presented in Table 2.

The structures of AG have different size and content. The example of the generated AG is in Table 2. Because the SOM training is random to some extent, the generated AG may vary between the particular network configurations. In most properly trained networks, some AG repeat, especially No. 1, 2 and 3. This proves the difficulty to distinguish between such categories. In most cases, each category is represented once in the group, which points at the limited degree of similarity between fault categories (as other examples of the same categories are not so similar and do not belong to this AG).

Table 1. Results of the SOM operation.

Topology	e_q	epochs	|AG|	WTM range
5x6	9.31e-2	25	6	1
8x10	8e-3	20	6	1
10x10	2.4e-2	30	7	1
15x10	6e-3	48	3	1
12x15	8e-3	65	2	1
15x16	3e-3	25	4	1
13x17	2.5e-3	25	4	2

Table 2. Contents of the AG.

AG No.	AG Content
1	11, 21
2	-31, 41
3	-11, -21
4	-12, -11, -21, -61
5	-22, -62

The produced result depends on the SOM training. If the weights of the network converge to the local minimum, the structure of the AG as in Table 2. The example of the correct quantization error e_q value change during the process is in Fig. 6. On the other hand, too large number of neurons or the incorrect value of SOM parameters (especially switching off the conscience mechanism) leads to the random-like behavior during the training process, as is presented in Fig. 7.

The training coefficient strongly influences obtained results. Combined with the initial range w_r of randomly selected weights determines the number of clusters generated on the examples from ***L***. The relation between the combination of these two parameters and knowledge about the data is illustrated in Table 3. Increase of η with relatively low value of w_r leads to the greater number of AG containing examples belonging to one or two fault categories. Also, the number of clusters $|c|$

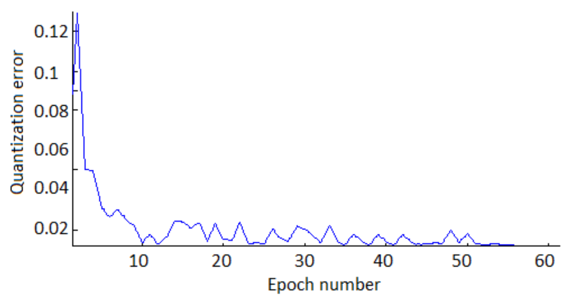

Fig. 6. Value of the quantization error during the SOM training.

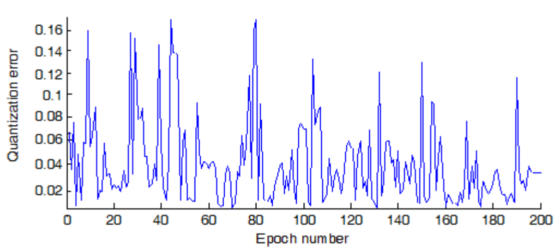

Fig. 7. The effect of the lack of convergence during the SOM training.

Table 3. Dependence between the training coefficient, range of initial SOM weights and results of the clustering.

η	w_r	WTM range	|AG|	|C|
0.3	0.3	1	5	31
0.3	0.2	1	5	30
0.7	0.2	1	4	36
0.3	0.1	1	9	28
0.7	0.1	1	7	31
0.3	0.1	2	9	27
0.5	0.1	2	10	23

The influence of the SOM size on the obtained structure of clusters is presented in Tab. 4. Here, only the rectangular configurations were tested, with increasing number of neurons in a row and all other parameters constant: the WTM range equal to 2, w_r*=0.3* and η*=0.4*. The number of clusters $|C|$ steadily increases with the increase of the number of neurons in the network. On the other hand, the number of ambiguity groups decreases, leading to the situation when most of examples belong to their own clusters (except the nominal state, grouped in one cluster). The analysis of the AG structure shows they contain in most cases two example. Clustering results show problems with the faults states "-11" and "-21", or "11" and "21". Selection of the topology the most representative for the particular SUT requires verification of the extracted information by the classification approach (such as the decision tree or the ANN) on the testing sets. It is expected that categories detected as AG would be difficult to distinguish, decreasing the accuracy of the classifier.

Table 4. Influence of the SOM size on the structure of the detected AG.

Topology	|C|	|AG|	e_q
25 (5x5)	17	11	0.20
36 (6x6)	21	9	0.14
49 (7x7)	27	9	0.10
64 (8x8)	28	7	0.11
81 (9x9)	32	9	0.07
100 (10x10)	35	6	0.04
121 (11x11)	35	5	0.03
144 (12x12)	34	5	0.04
169 (13x13)	38	3	0.03
196 (14x14)	41	2	0.01

The shape of the network plays a secondary role in the AG detection. Both square and rectangular configurations are able to converge and produce 3 to 5 groups, which may be difficult to distinguish for the classifier. The important issue is the adjustment of the SOM parameters to the particular topology. Also, various sizes should be tested to discover the change of the AG structures and their number.

VI. CONCLUSIONS

The method presented in the paper allows for the initial processing of the training data, detecting the potential problems for the AI-based diagnostic module. Application of the SOM for this purpose requires the careful adjustment of its parameters, including the topology and learning strategy and inclusion of the conscience mechanism. Obtained results give the general information about the difficulty of the data set and (indirectly) the analyzed SUT. To confirm the actual problems with distinguishing between categories, the classification algorithm must be trained on the learning set and then tested on the testing one. Efficiency of the particular classifier may depend on the form of knowledge extracted during the training. Therefore various classifiers may better or worse tackle the detected AG. This depends on the type of the algorithm used for the analysis. The SOM shows similarities between different examples based on their Euclidean distances, therefore the information about AG will be the most useful for classifiers operating on the same principle, such as ANN or k Nearest Neighbors (kNN).

The future investigations of the AG detection scheme should include introduction of additional clustering methods and their verification by the AI-based fault detection and location algorithms.

REFERENCES

[1] J.A. Starzyk, J. Pang, S. Manetti, M.C. Piccirilli, G. Fedi, "Finding ambiguity groups in low testability analog circuits," IEEE Trans. Circuits and Systems I: Fundamental Theory and Applications, Vol. 47, Issue 8, 2000, pp. 1125-1137.

[2] S. Manetti, M.C. Piccirilli, "A singular-value decomposition approach for ambiguity group determination in analog circuits," IEEE Trans. Circuits and Systems I Fundamental Theory and Applications 50(4): 477-487, May 2003.

[3] S. Khan, "Maintenance Requirements in Aerospace Systems," CIRP, Vol. 38, 2005 (Proceedings of the 4th International Conference on Through-life Engineering Services), pp. 100–105.

[4] S. Keck, E. Diecidue, D. Budescu, "Group Decisions under Ambiguity: Convergence to Neutrality," INSEAD Faculty & Research Working Paper, 2012, online: http://sites.insead.edu/facultyresearch/research/doc.cfm?did=49662

[5] J. Vesanto and E. Alhoniemi, "Clustering of the self-organizing map," IEEE Trans. Neural Networks, Vol. 11, Issue 3, May 2000, pp. 586 - 600.

[6] Y. Deng; Y. Shi; W. Zhang, "An Approach to Locate Parametric Faults in Nonlinear Analog Circuits," IEEE Trans. Instr. and Meas., Vol. 61, Issue 2, Feb. 2012, pp. 358-367.

[7] B. Cannas, A. Fanni, and A. Montisci, "Algebraic Approach to Ambiguity-Group Determination in Nonlinear Analog Circuits," IEEE Trans. Circuits and Systems—I, Vol. 57, No. 2, Feb. 2010, pp.438-447.

[8] A. Zymnis, S. Boyd, D. Gorinevsky, "Relaxed maximum a posteriori fault identification," Signal Processing 89 (2009), pp. 989–999.

[9] P. Bilski, „Application of6ty clustering method for the ambiguity groups detection in the diagnostic of analog systems," Przegląd Elektrotechniczny, No. 2a, 2013, pp. 276-278.

[10] P. Bilski, J. Wojciechowski, "Automated Diagnostics of Analog Systems Using Fuzzy Logic Approach", IEEE Trans. Instr. and Meas., Dec. 2007, Vol. 56, Issue 6, pp. 2175-2185.

[11] A. Bernieri, G. Betta, A. Pietrosanto, and C. Sansone, "A Neural Network Approach to Instrument Fault Detection and Isolation," IEEE Trans. Instr. and Meas., Vol. 44, No. 3, June 1995, pp. 747-750.

[12] G.N. Stenbakken, T.M. Souders, and G.W. Stewart, 1989, "Ambiguity groups and testability," IEEE Trans. Instr. and Meas., Vol. 38, Issue 5, 1989, pp. 941-947.

[13] N. Liu, J. Wang and Y. Gong, "Deep Self-Organizing Map for visual classification," International Joint Conference on Neural Networks (IJCNN), 12-17 July 2015, Killarney.

[14] J. M. Bossio, C. H. De Angelo, and G. R. Bossio, "Self-organizing map approach for classification of mechanical and rotor faults on induction motors," Neural Comput & Applic, No. 23, 2013, pp. 41–51.

[15] S.D. Sudhoff, B.T. Kuhn, K. Corzine, B.T. Branecky, "Magnetic equivalent circuit modeling of induction motors," IEEE Trans. Energy Conv., Vol. 22, No. 2, June 2007, pp. 259-270.

[16] K. Wang, J. Chiasson, M. Bodson, L.M. Tolbert, "An Online Rotor Time Constant Estimator for the Induction Machine," IEEE Trans. Contr. Syst. Techn., Vol. 15, No. 2, March, 2007, pp. 339-348.