Piotr BILSKI[1,2]

Warsaw University of Life Sciences (1), Warsaw University of Technology (2)

Application of clustering method for the ambiguity groups detection in the diagnostic of analog systems

***Abstract**. The paper presents application of unsupervised learning methods to detect ambiguity groups in the data used in the diagnostics of analog systems. The proposed approach processes labelled data sets from simulated systems to find similar examples belonging to different faulty states. Two algorithms were used in the presented research: graph clustering. Efficiency of the method is compared and verified against the exemplary electrical system, i.e. induction machine. Future prospects of such methods will be also included.*

***Streszczenie.** W artykule przedstawiono zastosowanie metod uczenia bez nadzoru w celu wykrycia grup niepewności w danych wykorzystywanych do diagnostyki systemów analogowych. Dane przetwarzane są w celu znalezienia podobnych do siebie przykładów należących do różnych kategorii uszkodzeń. Metoda clusteringu grafowego zostały przetestowane na przykładzie silnika indukcyjnego (**Zastosowanie metody grupowania w wykrywaniu niejednoznaczności w diagnostyce systemów analogowych**).*

Keywords: artificial intelligence, diagnostics of analog systems, unsupervised learning
Słowa kluczowe: sztuczna inteligencja, diagnostyka systemów analogowych, uczenie bez nadzoru

Introduction

Contemporary diagnostics of analog systems uses widely artificial intelligence methods. They require learning and testing data sets for extracting knowledge about characteristic features of each fault. The sets are obtained from simulation of system models (such as electronic circuits or electrical machines). The quality of the diagnostics depends on the information in data. Ambiguity groups, i.e. sets of system parameters indistinguishable based on the characteristic points (stamps or symptoms) obtained from the system's responses degrade the system's testability. Therefore they should be located and, if possible, eliminated [1]. They were considered especially in the analysis of circuits [2], but the problem applies to other objects as well.

The existence of ambiguity groups depends on the amount of information obtained from the system analysis. The most important factor is the number of available nodes at which measurements can be taken [2]. It is reflected by the size of the data set. Methodology of extracting useful information from the system responses is required. Characteristic points (symptoms or stamps) are obtained to maximize distinguishability between different system states.

The data set for the learning method is generated after introducing faults into the system model. It is excited with the selected signal and simulated. Stamps are collected at the accessible nodes. This way labelled examples are created (where the category of the fault or actual values of parameters are known). Examples belonging to different categories, but with similar stamps values exist. They are difficult to separate in the knowledge extraction process. Before training, the selected artificial intelligence method, such examples should be located.

Ambiguity groups are often present in the complex systems, where the access to all crucial sections is not possible (like in the feedback loop configuration). Distinguishing between problematic parameters requires increasing the number of the analysed nodes (sometimes, in integrated circuits, impossible) or applying multiple tests. Excitation signals are selected (step, sin or sinc functions), Domain of analysis (time, frequency or mixed) is chosen. Ambiguity group detection algorithms help in assessing the difficulty of the selected system and determine the testability.

The paper describes the application of two methods for unsupervised learning, where the data sets are processed disregarding labels of examples existing in data sets. Firstly, the problem is presented in detail. Next, the form of data sets used in diagnostics is explained. Then the algorithm for the ambiguity groups detection are introduced. Its application to the analysis of induction machine is presented (although it is possible to analyse any other technical system). Efficiency of method is evaluated for various parameter values. Finally, conclusions and future prospects for presented approach are explained.

Sources of knowledge about the analog system

The aim of the diagnostics is the discovery of the improper System Under Test (SUT) behaviour based on its responses. The model simulation helps in creating a set of examples, demonstrating SUT behaviour for various faults. Each time the selected fault is introduced, while all remaining parameters are nominal (within the tolerance margins). Extracting real-valued stamps (such as maximum values of the signal or its points of zero crossing) leads to creating a single example. These data are supplemented with the indicator of the fault source. The typical form of the data set *A* of *k* examples, each containing *l* stamps is in (1).

$$A=\begin{bmatrix} a_{11} & a_{12} & & a_{1l} & p_1 & v_{11} \\ a_{21} & a_{22} & & a_{2l} & p_1 & v_{12} \\ & & \cdots & & & \\ a_{k1} & a_{k2} & & a_{kl} & p_n & v_{nm} \end{bmatrix} \tag{1}$$

Here a_{ij} (*i=1,…,k*, *j=1,…,l*) are stamps values. The additional information is the number of the faulty parameter *p* and its value *v*. For the classification purposes fault codes are often used to easily indicate the particular categories. In the presented work they have two parts: one identifying the parameter and another one identifying the degree of deviation from the nominal value. For example, *-31* means that the third parameter has value smaller than the nominal, while *22* is for the second parameter having value much larger than the nominal. Thresholds between "larger and "much larger and "smaller and "much smaller were different (in both directions) at 30 and 60 percent from the nominal value, respectively. The latter has the code 0. This way every parameter of the SUT can be in one of five states. Changing their number affects the resolution of the classification method. The size of *A* depends on the number examples. Usually larger sets lead to better classification. The time of computations is the main limitation here.

Tested object

The work regime of asynchronous motors consists in rotating the rotor inside the changing magnetic field created by the current flowing through the stator. The typical model is described by the equations (2) [3]:

$$
\begin{aligned}
\frac{di_{sd}}{dt} &= \frac{\beta}{T_R}\cdot\varphi_{rd} + \beta\cdot n_p\cdot\omega_r\cdot\varphi_{rq} - \gamma\cdot i_{sd} + \frac{1}{\sigma\cdot L_s}\cdot u_{sd}\\
\frac{di_{sq}}{dt} &= \frac{\beta}{T_R}\cdot\varphi_{rq} - \beta\cdot n_p\cdot\omega_r\cdot\varphi_{rd} - \gamma\cdot i_{sq} + \frac{1}{\sigma\cdot L_s}\cdot u_{sq}\\
\frac{d\varphi_{rd}}{dt} &= -\frac{1}{T_R}\cdot\varphi_{rd} - n_p\cdot\omega_r\cdot\varphi_{rq} + \frac{M}{T_R}\cdot i_{sd}\\
\frac{d\varphi_{rq}}{dt} &= -\frac{1}{T_R}\cdot\varphi_{rq} - n_p\cdot\omega_r\cdot\varphi_{rd} + \frac{M}{T_R}\cdot i_{sq}\\
\frac{d\omega}{dt} &= \frac{M\cdot n_p}{J\cdot L_r}\cdot\left(i_{sq}\cdot\varphi_{rd} - i_{sd}\cdot\varphi_{rq}\right) - \frac{C_e}{J}
\end{aligned}
\tag{2}
$$

where electrical parameters are: 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 (3):

$$
\begin{aligned}
\beta &= \frac{M}{\sigma\cdot L_s\cdot L_r}\\
\gamma &= \frac{R_s}{\sigma\cdot L_s} + \frac{M^2\cdot R_r}{\sigma\cdot L_s\cdot L_r}
\end{aligned}
\tag{2}
$$

The model was simulated in the SIMULINK environment.

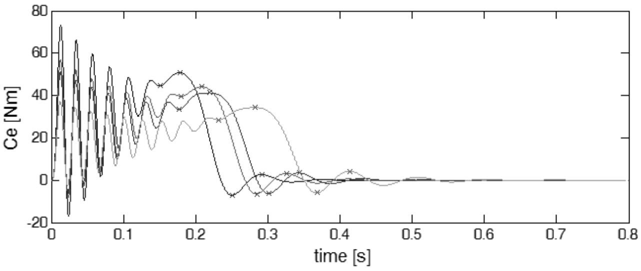

Fig. 1. Induction motor's torque for various values of R_s.

Seven analyzed parameters of the considered motor and their nominal values were R_s=2.25 Ω, L_s=0.1232 H, L_r=0.1122 H, M=0.1118 H, T_R=0.16 s, σ=0.09, J=0.0504. The remaining parameters remained unchanged at their nominal values throughout the test: R_r=0.7Ω, n_p=3. Each analyzed parameter was assigned seven values, leading to 49 total rows in the training and testing data set and 27 different fault categories. The exemplary response of the machine and the stamps extraction are in Fig. 1. Every analyzed fault is related to the particular physical effect, which might not be easy to isolate in the real machine.

Unsupervised learning method for ambiguity groups detection

The need to introduce unsupervised learning methods for the analysis of measurement data was explained in [4]. Artificial intelligence approaches process data sets searching for features to distinguish between different fault states. Unfortunately, some examples belonging to separate categories can have all values of stamps similar, making the distinction difficult or impossible. Such examples are ambiguity groups. In such a case the attempt to distinguish between these examples leads to the degradation of the diagnostic method (for instance, by producing incorrect rules). Therefore ambiguity groups must be identified. Locating them can be also used to find the minimal set of nodes, enabling the optimal classification efficiency. This would decrease the time and cost of measuring the real SUTs.

To detect ambiguity groups unsupervised methods can be applied. To use them, the categories of faults assigned during the simulation, must be disregarded. After grouping similar examples (creating clusters) it is possible to compare clusters with categories assigned by the designer. If the cluster contains examples belonging to different faults, the ambiguity group is suspected. Detecting it requires the data structure storing both original fault codes and the cluster identifier is required (Fig. 2).

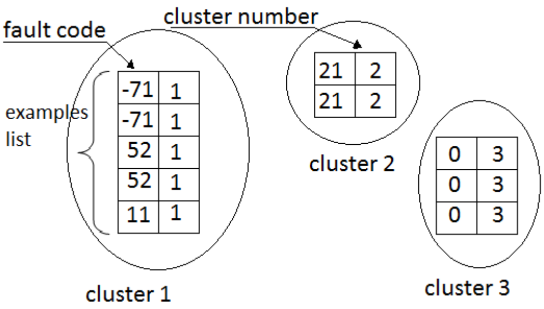

Fig. 2. Illustration of the clustering-based ambiguity group detection.

The clustering relies on the similarity measure. As stamps are real numbers, the most popular methods for this task are Euclidean and Manhattan distances, although there other are possible. The normalization of all stamps ranges is required, so they have identical influence on the overall distance. The key issue is determining the threshold, below which examples are considered similar and belonging to the same cluster. It determines the number of generated clusters. Different classes of algorithms are used: hierarchical, centroid-based or distribution-based [5]. In the first group examples are divided subsequently into subclusters until each belongs to its own category. The designer's task is to decide when this process should stop. In the second group the number of clusters is fixed and predefined. Each cluster is represented by one "pattern example. The third group is based on the statistical distributions of examples' stamp values. Recent development is aimed at working with large data sets. As multiple methods exist, it is reasonable to compare their outcomes and focus on differences.

For the presented experiments, the graph clustering algorithm was selected. Its detailed description is in [6]. Every example in the set *A* is treated as the vertex of the graph in *l*-dimensional space. Vertices are connected with edges if they are similar according to one of measures. The version of the algorithm used here was successfully applied to the geotechnical profiles generation [6]. Its similarity measure is as follows. The graph is created for every stamp separately, with individually adjusted similarity threshold. The resulting graph is a combination of all partial graphs and the edge between two vertices is present only if it exists between them in all partial graphs. Examples $\boldsymbol{a}_r$ and $\boldsymbol{a}_q$ must be then similar with respect to every stamp. This way there is no need to normalize their ranges.

The similarity threshold is calculated based on the histogram of the stamp's range. The stamp's values form clusters with empty spaces between them. The thresholds are calculated as the middle values in these empty areas. The important parameter is the number of intervals *σ* into which examples are divided to create the histogram. It was determined based on the number of examples in the processed set. Results of clustering for various numbers of intervals is in experimental section. This is the iterative process, as examples belonging to one cluster can also be partitioned using the histogram. It is terminated when the cluster contains examples belonging to only one category or after additional partitioning there is no further change in the cluster's structure. The exemplary histogram for the maximum value of torque C_e is in Fig.3. Four thresholds *θ* and five clusters are visible here. The cluster No. 2 is the most promising for further partitioning.

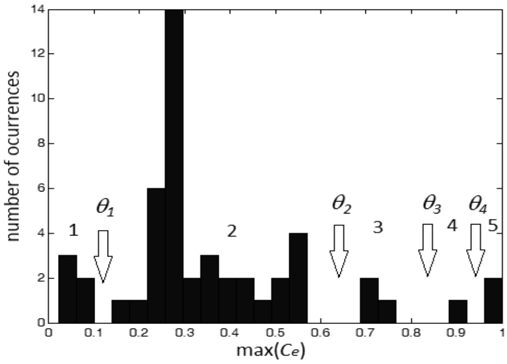

Fig. 3.Histogram for the maximum value of the torque stamp

The recurrent process of generating clusters is in Fig. 4. Here, *C* is the set of clusters, *K* is the processed column (stamp) from *A*, ***θ*** is the set of thresholds obtained by the get_histogram method, ***L*** is the set of partitioned examples based on theresholds, *C(**L**[i])* is the set of clusters obtained from dividing the *i-th* cluster into subclusters. After generating |***L***| clusters, each is verified if further partitioning is not needed (function (complex)). If so, ***L**[i]* examples are recursively processed.

```
create_clusters(K, σ)
C ← Ø
    θ ← get_histogram(K,σ)
    L ← divide_examples(K, θ)
    for i=1 to |L|
        if complex(L[i])
                σ ← get_intervals(L[i])
                C(L[i]) ← create_clusters(K(L[i]), σ)
                C ← C + C(L[i])
        else
                C ← C + L[i]
        end
    end
return C
```

Fig. 4.Algorithm for the recurrent clusters generation

Test results

Performed experiments were aimed at verifying the ability of the method to detect ambiguity groups. Results for three values of *σ* are in Tab 1, where *n* is the coefficient dividing the number of examples to obtain *σ*. Increase of the intervals number allows for eliminating the largest cluster, which involves distinguishing the fault-free state from small deviations of numerous parameters. Further increase of intervals leads to decreasing the number of clusters, eventually leaving two ambiguity groups: *{0, -71}* and *{-11, -51}*. This is possible when most of examples belong to their own separate categories. Verification of ambiguity groups using rough sets classification method showed that clusters for *n*=1.5 are difficult to distinguish, therefore this value is the most accurate for this SUT.

Table 1. Results of ambiguity groups detection with graph clustering algorithm

n	*σ*	*Cluster 1*	*Cluster 2*	*Cluster 3*
2	25	*-11 0 -21 -31 41 -51 -61 61 -71 71*	*-11 -51*	*11 21 51 -71*
1.5	33	*0 -51 -71*	*-11 -21 41*	*11 21*
1.3	38	*0 -71*	*-11 -51*	*X*
1.2	41	*0 -71*	*X*	*X*

Summary

The discussed approach allowed to analyse the SUT according the corresponding data set. Ambiguity groups identified on the appropriate level of set partitioning cause trouble for the rough sets-based diagnostic system, which proves their usefulness. Some clusters for too small *σ* were not problematic, which raises the question about the optimal resolution of the approach. As there are more unsupervised learning methods, they should also be tested to compare obtained clusters and select the best approach for the task. Also, the method for selecting the minimal set of stamps by minimizing the number of ambiguity groups should be investigated in the future.

REFERENCES

[1] S. Manetti, M.C. Piccirilli, A singular-value decomposition approach for ambiguity group determination in analog circuits, IEEE Trans. CAS I: Fundamental Theory and Applications, Vol. 50, Issue: 4 (2003), pp. 477-487.

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

[3] 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 (2007), pp. 339-348.

[4] P. Bilski, "An unsupervised learning method for comparing the quality of the soft computing algorithms in analog systems diagnostics, Electrical Review, No. 11a (2010), pp. 242-247.

[5] H.-P. Kriegel, P. Kröger, J. Sander, A. Zimek, Density-based Clustering. WIREs Data Mining and Knowledge Discovery, Vol. 1, No. 3, (2011), pp. 231–240, doi:10.1002/widm.30.

[6] P. Bilski, S. Rabarijoely, "Automated soil categorization using the CPT and DMT investigations, Proc. 2nd Int. Conf. on New Developments in Soil Mechanics and Geotechnical Engineering ZM2009, Nicosia, Northern Cyprus , May 28-30 2009, pp. 368-375.

***Authors**: dr inż. Piotr Bilski, Szkoła Główna Gospodarstwa Wiejskiego, Katedra Zastosowań Informatyki, ul. Nowoursynowska 159, 02-776 Warszawa, E-mail: piotr_bilski@sggw.pl; Politechnika Warszawska, Instytut Radioelektroniki, ul. Nowowiejska 15/19, 00-665 Warszawa, E-mail: pbilski@ire.pw.edu.pl.*