*Int. J. Appl. Math. Comput. Sci.*, 2016, Vol. 26, No. 3, 655–668
DOI: 10.1515/amcs-2016-0045

# AUTOMATIC PARAMETRIC FAULT DETECTION IN COMPLEX ANALOG SYSTEMS BASED ON A METHOD OF MINIMUM NODE SELECTION

ADRIAN BILSKI [a,*], JACEK WOJCIECHOWSKI [b]

[a] Faculty of Applied Informatics and Mathematics
Warsaw University of Life Sciences—SGGW, ul. Nowoursynowska 159, 02-776 Warsaw, Poland
e-mail: adrian_bilski@sggw.pl

[b] Institute of Radioelectronics
Warsaw University of Technology, ul. Nowowiejska 15/19, 00-665 Warsaw, Poland

The aim of this paper is to introduce a strategy to find a minimal set of test nodes for diagnostics of complex analog systems with single parametric faults using the support vector machine (SVM) classifier as a fault locator. The results of diagnostics of a video amplifier and a low-pass filter using tabu search along with genetic algorithms (GAs) as node selectors in conjunction with the SVM fault classifier are presented. General principles of the diagnostic procedure are first introduced, and then the proposed approach is discussed in detail. Diagnostic results confirm the usefulness of the method and its computational requirements. Conclusions on its wider applicability are provided as well.

**Keywords:** complex analog systems, support vector machine, tabu search, genetic algorithm, parametric fault detection.

## 1. Introduction

The continuously increasing number of elements in analog systems makes their testability and diagnostics difficult. In the case of systems working in high frequencies or performing the data acquisition, it is necessary to conduct separate diagnostics for analog and digital parts. The diagnostics of the latter has well established testing procedures, while analog or mixed systems are more difficult to analyze in a uniform manner. The testability of all types of circuits is crucial to decrease the costs of production in modern electronics (Milor, 1998; Huertas, 1993). Fault diagnostics in analog systems is also complicated because tolerances of elements must be taken into account.

When a new analog circuit is introduced, diagnostic and testing methods must be designed for it. Early detection of design faults allows decreasing production costs (Milor, 1998; Bushell and Agrawal, 2002). In the case of digital circuits, the commonly known IEEE1149.1 norm is applied. The testability of mixed systems has been described by Bushell and Agrawal (2002) (the IEEE 1149.4 norm). No such unified approach exists for analog systems at the moment.

The aim of the diagnostic process is to assess whether the system under test (SUT) is functioning according to the design specifications, based on the analysis of some observable functions $f(\mathbf{x}, t)$ recorded at accessible or partially accessible nodes. Here $\mathbf{x}$ denotes the vector of parameters (usually real numbers), while $t$ is time. The analysis is performed based on the information (characteristic points) taken from $f(\mathbf{x}, t)$ and consists in determining how much they differ from the nominal values of symptoms.

An accessible node is a node available for the measurement/control equipment. They are usually identified as external nodes, such as input signal nodes (power supply included), output signal nodes and selected internal nodes of the analyzed system. Two different testing approaches can be distinguished (Milor, 1998):

- specification-driven test (SDT), which is used to assess whether the system meets the specification,
- fault-driven test (FDT), which is used to find the faulty element responsible for the system's faulty behavior.

The main goal of diagnostics is fault detection and

[*] Corresponding author

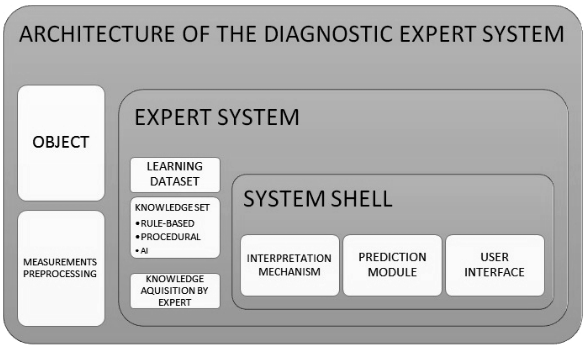

Fig. 1. Architecture of a diagnostic expert system (Korbicz *et al.*, 1994).

location. Reaching both of them depends on the quality of the methods applied, physical SUT characteristics and the number of accessible or partially accessible nodes. For example, the faulty element being part of the feedback loop might not be easily detectable.

Proper selection of characteristic points (symptoms, extracted from the SUT response signals) creates the possibility of detecting and locating the fault. Through parametric simulation it is possible to acquire information about the behavior of a complex analog system by changing the values of its parameters and observing responses. Such methods are similar to the value limit control method checking whether the selected characteristic point $V$ belongs to the set of values ensuring the proper operation of the system,

$$V_{\min} \leq V(t) \leq V_{\max}. \tag{1}$$

Though some efforts to formulate the scheme for the diagnostics of complex analog systems have been made, for example, by decomposing such objects into a number of connected simpler circuits, they do not provide any effective and generic methodology.

Accurate diagnostics of complex analog systems requires optimization of the set of accessible nodes and the selection of excitation signals. The probability of effective fault identification significantly decreases with an increase in the size of the diagnosed system. This justifies searching for more effective methods in this field.

For the purpose of this paper, we use a diagnostic expert system (Fig. 1), exploiting one of two binary classifiers (the support vector machine and the $k$ nearest neighbors, $k$NN) as fault detectors and two discrete optimization schemes (tabu search and a genetic algorithm) as node selectors. The SVM is able to learn from data, extracting knowledge about the particular faults. It also provides good results in measurement uncertainty conditions. The $k$NN algorithm is used as an alternative classification method for comparison purposes.

The following paper introduces a methodology for selecting the minimum number of nodes in a complex analog system, providing sufficient accuracy of diagnostics (better than for the input-output analysis). The diagnosed objects are limited to linear systems with the number of parameters large enough to make them difficult to analyze based on just the input-output characteristics.

In the input-output analysis the knowledge about the source of the fault is collected only from the output node while using the input node for excitation. This is justified, since modern analog circuits are mostly integrated in a single chip, which limits the set of possible accessible nodes.

The paper is organized as follows. Section 2 presents the diagnostic principles. A description of the data processing method and the minimal node selection algorithms are in Section 3. Section 4 introduces the analyzed SUT examples, i.e., a 26-element video amplifier and a 52-element low-pass filter. In Section 5 experimental results are presented, while Section 6 contains the evaluation of the implemented artificial intelligence (AI) methods and their perspectives in the presented domain.

## 2. Diagnostic principles

Access to internal nodes of integrated circuits is limited. For both the FDT and SDT, two different techniques are possible to apply: simulation before test (SBT) and simulation after test (SAT) (Chakrabarti *et al.*, 1999). The first one concerns the type of simulation where calculations are conducted using the system's model, before the actual object is analyzed. The second one focuses on computations during measurements from the actual system. One of significant differences between them is the extensiveness of calculations that need to be conducted after the measurement data are collected from the actual system. This feature determines whether the diagnostic method can be used in real-time mode.

A different method of fault searching in analog systems is the built-in self test (BIST). Originally designed for complex digital chips, now it serves diagnostic purposes for many devices with a relatively small amount of digital components. With the advent of advanced line process geometries, the BIST functionality has been included in analog data converters. This autonomous testing procedure can simplify the chip manufacturing process by providing greater visibility into the diagnosed device and enhancing its reliability while reducing the production time. The basic idea behind this method is to add the subsystem to the integrated circuit or the printed circuit board to make it test itself. At the end of the BIST sequence, the pass/fail result report is acquired.

Early implementations of the BIST for the analog part of integrated systems required elements typically used for digital systems testing. This included a test pattern generator (TPG), an output response analyzer

(ORA) and a test controller (TC). The TPG stimulates the SUT, while the response analyzer gathers test patterns compressing it to a single signature. The test controller is the diagnostic process coordinator, providing an external interface. The TPG usually contained linear feedback shift registers (LFSRs), which generated pseudorandom digital patterns resembling white noise, when passed through a digital-to-analog converter (DAC) (Ohletz, 1991; Pan and Cheng, 2000). This waveform was considered by some to be a universal tool for analog systems testing. Currently ramp signals are utilized for testing purposes of analog parts, becoming a significant tool for proper diagnostics and in some cases better than sine wave test signals (Chatterjee *et al.*, 1996).

One of the main problems in diagnostics is testability (Huertas, 1993), which relies on optimal test point selection (Prasad and Babu, 2000; Starzyk *et al.*, 2004; Golonek and Rutkowski, 2007), including stimulus parameters and features of the output signals (Grzechca *et al.*, 2007; Grasso *et al.*, 2007).

For test point optimization, the entropy measure (Starzyk *et al.*, 2004) and the genetic algorithm (Golonek and Rutkowski, 2007) have been used. The optimization of the input signal is also performed: in the time (Golonek *et al.*, 2008) and the frequency domain (Sen and Saeks, 1979; Grzechca *et al.*, 2007). Grasso *et al.* (2007) used the simulated annealing algorithm in optimizing multitonal signal selection.

Many AI algorithms have been applied as fault classifiers (Aminian and Modular, 2007; Jantos *et al.*, 2009), delivering better classification results than traditional methods (Hochwald and Bastian, 1979). In the works of Aminian and Modular (2007) as well as Kuczyński and Ossowski (2009), the wavelet transform in the preprocessing stage was introduced, improving fault classification in electronic systems. The artificial neural network (ANN) was utilized for single parametric fault classification (Jantos *et al.*, 2009). A different approach to fault detection was introduced by Czaja and Zielonko (2004), who transformed measurements into a multidimensional space. In the work of Tadeusiewicz and Hałgas (2006), an algorithm for multiple catastrophic fault was proposed, requiring accurate measurements at test points. The taxonomy of AI methods used for the purposes of analog systems diagnostics is presented in Fig. 2.

Though generic methods for effective diagnostics of complex analog systems do not exist, some attempts at creating them have been made (Bilski, 2007; Tadeusiewicz and Korzybski, 2000; Rutkowski and Grzechca, 2009). An effective approach is the decomposition method (Starzyk and Dai, 1992; Salama *et al.*, 1984), based on creating test Kirchhoff current equations at the circuit decomposition nodes. Significant computation times limited the size of possible circuits, and such methods are now considered to be obsolete with the advent of modern heuristic approaches.

The SVM classifier has been used in diagnostics of electronic circuits (Tadeusiewicz and Korzybski, 2000; Rutkowski and Grzechca, 2009; Guo *et al.*, 2014; Sun *et al.*, 2013; Bilski, 2011; Sałat and Osowski, 2011). Although this tool is versatile to work with data of various complexities, the process of optimal kernel selection and its parameters must be conducted for each system separately (Bilski, 2011). It also provides sufficient classification results in noisy conditions. The objects presented by Rutkowski and Grzechca (2009), Guo *et al.* (2014) or Sun *et al.* (2013) are characterized by a small number of parameters (from 7 to 16 elements). The SVM presented there is used for classification purposes in fault detection. Such circuits are of significantly lesser complexity than the system analyzed in this work. The benchmark complex circuit used for testing diagnostics algorithm is the cascade of Sallen–Key sections.

In the time domain analysis, fault classification is based on selected symptoms of the system's response (Golonek *et al.*, 2008). A fault dictionary is then built (Hochwald and Bastian, 1979). To increase classification efficiency, a variety of different soft computing algorithms are used (Grzechca *et al.*, 2006; Bilski and Wojciechowski, 2007; Wang and Yang, 2005). To detect ambiguity groups, a Monte Carlo analysis has been used (Huang *et al.*, 2010), though it is not computationally efficient. Fuzzy logic and rough sets were used in diagnostic systems to increase the number of inputs equal to the number of diagnosed parameters (Bilski and Wojciechowski, 2012). The disadvantage of this algorithm is the absence of a module for extracting knowledge from the given datasets. Fuzzy logic does not allow automated knowledge extraction—it has to be provided by an expert or an external machine learning method. The main problem with rough sets is the dependence of results on the discretization method.

Rough sets are also rule-based and they are adequate in finding dependencies in data. In the second case the decision about the system's state is based on the voting process. To transform characteristic points and make them more easily distinguishable, a statistical method can be used (i.e., PCA) (cf. Nguyen and Golinval, 2010).

Most diagnostic procedures focus on single fault detection, assuming that only one parameter is simultaneously beyond the tolerance range (Fang *et al.*, 2006; Spina, and Upadhyaya, 1997). It is the most probable situation, although multiple fault occurrence is possible in actual scenarios. A method for multiple catastrophic fault detection has been proposed (Tadeusiewicz *et al.*, 2011). Such methods do not exist for parametric faults.

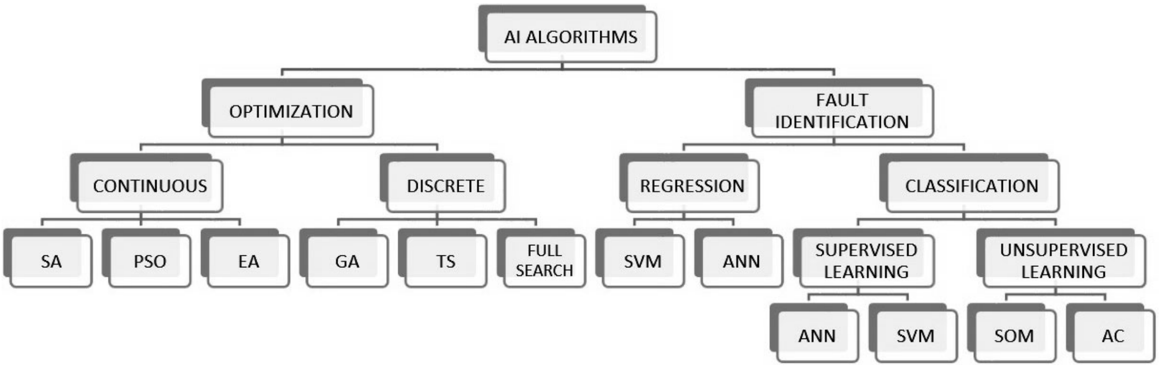

Fig. 2. Classification of intelligent methods used in diagnostics.

Table 1. Dataset of examples.

| P1 | T1 | P2 | T2 | Element code | Element parameter value | Fault code |
|---|---|---|---|---|---|---|
| 1.46 e−001 | 3.23 e−006 | −1.46 e−001 | 3.73 e−006 | 3.00 e+0 | 2.50e+001 | 0 |
| 2.22 e−002 | 3.23 e−006 | −2.24 e−002 | 3.73 e−006 | 1.00 e+0 | 7.00e+002 | 1 |
| 3.92 e−002 | 4.22 e−006 | −3.90 e−002 | 4.72 e−006 | 1.00 e+0 | 5.00e+002 | 1 |
| 5.44 e−002 | 4.22 e−006 | −5.19 e−002 | 4.70 e−006 | 4.00 e+0 | 1.00e+002 | 1 |

## 3. Methodology applied

**3.1. Creation of fault datasets.** Dataset creation is a standardized data-driven procedure. The problems that need to be solved here are the size of the dataset and differentiation on learning and training datasets.

The latter can be solved by partitioning datasets into two sets of equal size or by cross validation. Here the first option is utilized, providing the same amount of data to learn from and to test the algorithm.

Data classification based on AI algorithms assigns measurable information to a certain fault identifier. To achieve this, knowledge about the relation between the characteristic points and fault categories is required. The learning set contains examples describing the system's behavior in various faulty situations. To verify the quality of knowledge extracted by the machine learning method, a validating set is required. All sets are created after the simulation of the SUT model. The additional validating set is often used during parameter optimization of heuristic fault classifiers (such as the SVM). The testing dataset has the form similar to the training and validating ones, but contains examples other than in the remaining sets.

Each dataset is a table in which every row (example) contains values of symptoms, supplemented by a discrete identifier referring to the actual value of the selected system parameter. To differentiate the degree and direction of the parameter deviation from its nominal value, unique codes are assigned to the particular elements' values. Multiple simulations are performed to create required examples. After changing the value of the selected parameter, the response of the SUT is recorded. The procedure is repeated for every parameter separately (the remaining ones are at nominal values). The parameters are subsequently assigned multiple values (the so-called parametric sweep). The fault-free state is also simulated and added to the data set. Table 1 shows a part of the training dataset created for the 52-element low-pass filter (Fig. 10). Here ($P_1$,$T_1$) and ($P_2$,$T_2$) are the coordinates of the first two extreme values of the input signal. The element code is the number of the circuit element whose value is currently changed. It is used to assess whether or not the diagnosed system is faulty.

In the presented experiment, each SUT element is described by eight examples (being sets of characteristic points acquired by parametric simulation during which only one parameter is susceptible to a change). Some elements have low sensitivity and may require a greater deviation from the nominal value than others to make the change visible in the output signal.

**3.2. Fault classification tools.** The SVM classifier is an equivalent of a single perceptron (Widodo and Bo-Suk, 2007), thus some researchers regard it as a type of ANN. Its purpose is to effectively separate examples in the space of kernel functions. Their parameters are optimized to maximize the generalization of knowledge. The main difference between SVMs and other ANNs is the lack of local minima in the target function. The training of a linear SVM classifier for the diagnostic task can be formulated in the following way (Osowski, 2006).

Given a learning set $\mathbf{X}$ (represented by $n$ examples: $(\mathbf{x}_1, \dots, \mathbf{x}_n)$) and the corresponding labels $y_i \in \{-1, +1\}$, $i = 1, 2, \dots, n$, a vector $\mathbf{x}$ can be associated with one of two categories. The classification function is described in the following way:

$$\begin{cases} \text{if} \quad \mathbf{w}^T\mathbf{x}_i + b > 0, & \text{then} \quad y_i = +1, \\ \text{if} \quad \mathbf{w}^T\mathbf{x}_i + b < 0, & \text{then} \quad y_i = -1, \end{cases} \tag{2}$$

$$\mathbf{w}^T \cdot \mathbf{x} + b = g(\mathbf{x}), \tag{3}$$

where (3) defines a hyperplane separating two different categories; $\mathbf{w}$ is a vector whose length determines the distance between the hyperplane and the nearest incorrectly classified example, while $b$ is a scalar determining the hyperplane's position (Fig. 3).

To optimize the SVM classifier, the analysis of different kernel functions is necessary. Most of them are described by nonnegative real values (like the width of RBFs or the degree of the polynomial), which affect the diagnostic process. The most popular SVM kernels with their proper parameters can be found in Table 2.

In the case of $k$NN classification, each testing example is compared with the whole training set using the selected metric. A particular class is assigned to the analyzed example as a result of majority voting of its nearest $k$ neighbors, so that the closer ones contribute more to the average than the distant ones.

The parameter $k$ determines how many neighbors take part in the voting process. Each neighbor is weighted to differentiate the more meaningful ones from the rest. Here a weight of $1/d$ is used, where $d$ is the distance between the analyzed example and its particular neighbor. This algorithm can be perceived as an extension of the classical dictionary method.

Here $k$NN classification performance can also be improved through proper metric selection. From multiple potential metrics proper for calculating differences between objects in the $m$-dimensional space, i.e., Manhattan/cityblock, Euclidean, cosine and Chebyshev, the first two were selected. The distance values between given examples calculated using these metrics best fit the fault scheme. Table 3 presents different distance metrics, where $a_s$ and $b_t$ are two vectors of examples between which the distance is calculated.

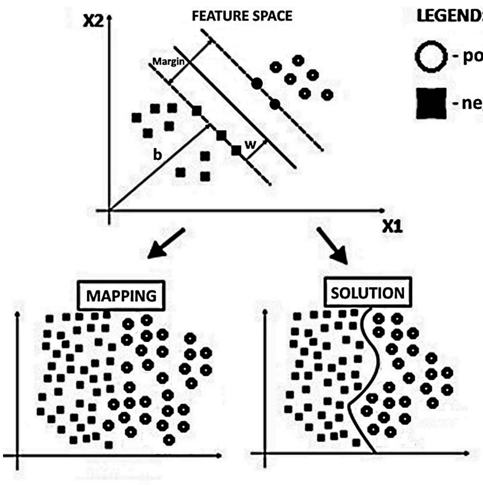

Fig. 3. Separation of two classes using the SVM (Osowski, 2006).

Table 2. Types of kernels.

| Kernel type | Parameter | Equation |
|---|---|---|
| RBF | Width of RBFs | $\exp(-\gamma\|\|x - x_i\|\|^2)$ |
| polynomial | Degree | $K(x, x_i) = (x^T x + \gamma)^p$ |
| sigmoid | Scale and offset | $\tanh(\beta_i x^T x_i + \beta_0)$ |
| linear | None | $K(x, x_i) = x^T x + \gamma$ |

Table 3. Types of distance metrics.

| Metric type | Equation |
|---|---|
| Manhattan (cityblock) | $d = \sum_{j=1}^{n} \|a_{sj} - b_{tj}\|$ |
| Euclidean | $d = \sqrt{\sum_{j=1}^{n} \|a_{sj} - b_{tj}\|^2}$ |
| Chebyshev | $d = \max_j \{\|a_{sj} - b_{tj}\|\}$ |
| cosine | $d = 1 - \frac{a_s b_t'}{\sqrt{(a_s a_s')(b_t b_t')}}$ |

Table 4. Example index matrix created for the $k$NN classification algorithm.

| Example | NN1 | NN2 | NN3 | NN4 |
|---|---|---|---|---|
| 1 | 4 | 2 | 5 | 1 |
| 2 | 5 | 1 | 3 | 12 |
| 3 | 1 | 3 | 12 | 10 |
| 4 | 5 | 1 | 3 | 12 |
| 5 | 9 | 11 | 10 | 3 |
| 6 | 4 | 6 | 2 | 5 |

During the process of classification, the $k$NN algorithm produces the index and distance matrix; cf. Tables 4 and 5, respectively. The index matrix contains examples from the training set (neighbors' positions) closest to the analyzed one. The size of the matrix depends on the number of arbitrarily defined $k$ neighbors. To avoid the case when the resulting categories are supported in the voting process by an identical number of examples (leading to a draw), only uneven values of $k$ were considered. Table 5 presents an example of different distance values calculated for the same neighbors using all the four metrics.

Table 5. Example of different distances calculated for the same neighbors using various metrics.

| $k$ | Euclidean | Manhattan | Cosine | Chebyshev |
|---|---|---|---|---|
| 1 | 0.01745 | 0.021334 | 0.13639 | 0.07377 |
| 2 | 0.00715 | 0.288083 | 0.21012 | 0.02770 |
| 3 | 0.15504 | 0.036855 | 0.24869 | 0.00449 |
| 4 | 0.48315 | 0.355986 | 0.68764 | 0.08667 |
| 5 | 0.16411 | 0.176288 | 0.16135 | 0.05021 |

**3.3. Node selection.** If the diagnostic mechanism is automatic and the analyzed object is not too complex, all system nodes may be considered in the experiment. To decrease the computational cost, a reduction in the node set size is performed by rejecting analytically irrelevant nodes. So far this has required an expert's knowledge and was not performed automatically. The idea proposed in this paper is to minimize the impact of the human expert on the diagnostic process, including the selection of the analysis domain, excitation signal parameters or test nodes.

While making a decision about the nodes included in the analysis, discrete optimization algorithms were applied for node selection. Although multiple approaches are possible here, two heuristic representatives, i.e., tabu search and the GA, were selected. The first one is simple but powerful, while the second is used as the "last chance method" when all the others fail. It was exploited as a reference for the tabu search.

Both methods work assuming that all nodes in the analyzed system can be accessible (excitations put to them and responses read from). A successful solution presents nodes that should be excited and measured to get better results than the input-output analysis, working in the shortest possible time. Such information can then be used by the designer to make the diagnostics easier and cheaper. This method is important from the designer's point of view, as it demonstrates which nodes should be made accessible in the pins of the integrated circuit. This is the cheapest way to ensure high testability of the actual circuit.

The target function of the proposed algorithms is

$$f = \frac{f_e(T)}{v\dfrac{h_a}{h} + (1-v)\dfrac{t_a}{t_{\max}}}, \tag{4}$$

where $h_a$ is the number of accessible nodes considered in the analysis, $h$ denotes the total number of all nodes in the circuit, $t_a$ is the optimization time reached by a particular solution, while $t_{\max}$ is the maximum permitted time of the training and testing operation for the full set of symptoms. The weight $v$ belongs to the range of 0–1. It is decided based on this value whether the emphasis of the optimization process is put on the minimization of the number of nodes or the time of conducted calculations. Because it is assumed that the minimization of the number of accessible nodes is more important, which leads to simpler and faster object analysis, the weight value is set to $v = 0.8$.

The classification efficiency for the training set $f_e(T)$ is the percentage of correctly categorized examples ($p_0$) from all analyzed by the method ($n$),

$$f_e(T) = \frac{p_0}{n}. \tag{5}$$

The optimization mechanism introduced in this paper can be summarized up as follows:

- a node configuration is randomly selected;
- for this configuration, training and testing datasets are created;
- diagnostic quality is obtained by learning and testing the fault classifier;
- a new solution is created, for which the whole procedure is repeated.

The given process requires approximate methods as it is very time consuming. For a solution created from 12 nodes, $2^{12} = 2048$ different combinations have to be considered. In this case the use of an evolutionary algorithm as a node selector can be very costly, as it produces multiple solutions at the same time while the training/testing of the fault classifier has to be conducted for each one. Thus the evolutionary algorithm fits better a parallel implementation. The diagnostic process is subsequently elongated by the fact that the SVM kernel parameter is optimized for each node configuration. Table 6 contains optimal values of the evaluation function for optimal node selection.

The algorithm operates on sets of examples that were already created for each node before the optimization

Table 6. Target function values for optimal node selection.

| Algorithm | Target (GA) | Target (tabu) |
|---|---|---|
| 26-element systems analysis | | |
| SVM (RBF) | 8.87e−4 | 0.155 |
| SVM (ERBF) | 8.87e−4 | 0.156 |
| SVM (poly) | 7.6e−4 | 0.124 |
| $k$NN (cityblock) | 7.6e−4 | 0.124 |
| $k$NN (Euclidean) | 7.6e−4 | 0.124 |
| 52-element systems analysis | | |
| SVM (RBF) | 3.931e−4 | 2.65 |
| SVM (ERBF) | 3.931e−4 | 2.65 |
| SVM (poly) | 3.528e−4 | 2.38 |
| $k$NN (cityblock) | 3.357e−4 | 2.38 |
| $k$NN (Euclidean) | 3.357e−4 | 2.38 |

started. Such an approach significantly speeds up the algorithm. To limit the number of generated solutions, nodes located near the power supply can be eliminated. Because experiments presented here are based on the AC analysis, it can be assumed that the system's faulty behavior cannot be caused by a faulty power supply, as these types of faults are impossible to detect.

The tabu search algorithm is iterative and it works as in Algorithm 1 (Gendreau, 2003). Solution vectors are binary with the length equal to the number of nodes in the SUT: value 1 is assigned to the node selected for testing. The successive solutions are created from the latter ones by choosing a different tabu element, thus including or excluding a single node from the previously generated solution (the Hamming distance is 1). The new element remains tabu for a number of iterations depending on half the number of nodes in the analyzed system. The memory horizon is defined by the short term list. The new solution is a different node configuration. The stopping criterion for tabu search was reaching 100 iterations with no improvement of the evaluation function. This condition is introduced to limit the duration of a algorithm execution. Based on Fig. 4 it can be shown that the better solution than the one achieved by the input-output only solution can be achieved within the first 50 iterations. It is also the main difference between this algorithm and the full sweep method, which checks all combinations of nodes and in theory provides the best classification results.

**Algorithm 1.** Tabu search procedure.

```
Solution=best=random solution
Tabu list:= 0
While not (stop condition) do
    Begin
        Evaluate best
        Solution=select(neighbors(solution),tabu list)
        Tabu list=update tabu(solution,tabu list)
        Evaluate Solution
        If (evaluate(best); evaluate(solution))
        Best=solution
    End
```

Node selection using the genetic algorithm (GA) was adopted according to the scheme proposed by Holland (see Algorithm 2). The initial population $P_0$ is selected randomly. Each member of the population is a binary vector of the same form as in tabu search. The population consists of 60 solutions (vectors), which is enough to maintain its diversity. During the evaluation of a population (the assessment of which solution provides the best classification results), a particular solution is selected for reproduction based on fault classification performance, provided by the SVM classifier (5). Genotypes of the selected solutions (vectors of ones and zeros, which hold the information which nodes are considered in a particular simulation) undergo evolutionary operations, with the probability of crossover ($p_c$) equal to 0.7 and mutation ($p_m$) equal to 0.01. These are the values that provide the highest variety in the reproduction process of populations, with the crossover operation more significant in these tasks.

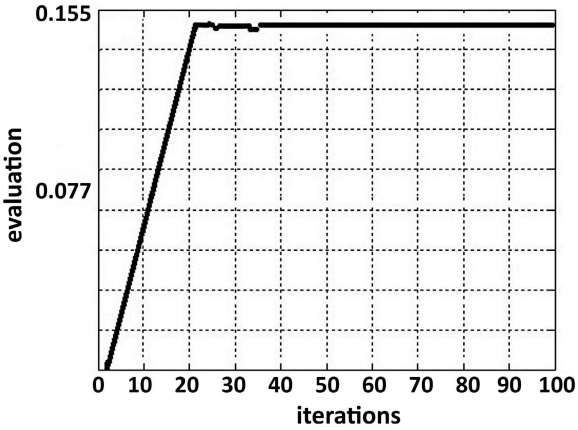

Fig. 4. Tabu search evaluation function.

**Algorithm 2.** Basic GA procedure.

```
$t := 0$
Initialization of $P_0$
Evaluation of $P_0$
While not (stop condition) do
    Begin
        $T_t$ := reproduction of $P_t$
        $O_t$ := crossover and mutation of $T_t$
        Evaluation of $O_t$
        $P_t + 1 := O_t$
        $t := t + 1$
    End
```

Figure 6 demonstrates changes in the probabilities of the crossover (the dashed line) and mutation during the process of their optimization in relation to classification efficiency. The intersection of these lines constitutes the optimal values of $p_c$ and $p_m$. The method of solution selection used in this study is proportional reproduction described by

$$p_r(x) = \frac{\phi(x)}{\sum\limits_{y \in p_t} \phi(y)}, \tag{6}$$

where $x$ is a single solution vector while $\phi(x)$ is the value of its evaluation function. The cross over method used in these studies is single-point (see Fig. 5).

The GA stopping criterion was generation of the 100th population without the improvement of the evaluation function.

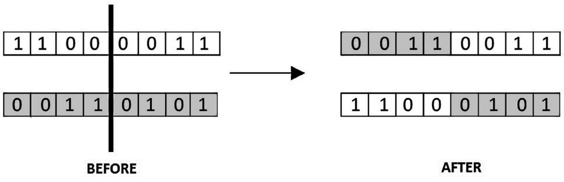

Fig. 5. One-point crossover.

As shown in Fig. 7 this is a sufficient number of populations to achieve a better solution than the one provided by the input-output analysis. The GA quickly converges within the first 30 generated populations. Results of the algorithm depend on its parameters, although it is of a stochastic nature (the selection of the first solution and its derivatives). The most notable effect is that by the mutation probability on the evaluation divergence of a particular population. A significant value of $p_m$ causes substantial divergence in subsequent populations, while its low value causes the populations to be more uniform. The crossover probability influences the scope of the solution search of the algorithm. For low $p_c$ values, the algorithm will not generate all possible solutions, even for a large number of iterations.

**3.4. Diagnostic method.** This paper focuses on single parametric fault detection. The information about the SUT behavior is acquired from the signals provided by the parametric simulation (Fig. 8) of faulty and nominal systems. Examples for the testing dataset were selected randomly from the initial set of measurements. Element tolerances were considered in the experiment, randomly changing in the range of $+/-5$ around the nominal value. It is assumed that the system works correctly if the parameter values do not exceed their tolerances. The values of the outputs acquired when the analyzed element parameters reach their tolerance values are also included in the datasets. Two types of fault detection are considered; binary, i.e., go/no-go (system operates properly/improperly), and more precise, i.e., parameter identification (the value of a parameter is smaller, larger, much smaller, much larger than the nominal value, and so on).

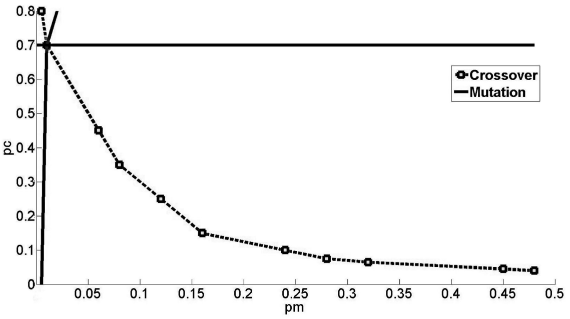

Fig. 6. Illustration of the mutual dependence between the crossover and mutation probabilities regarding the classification accuracy.

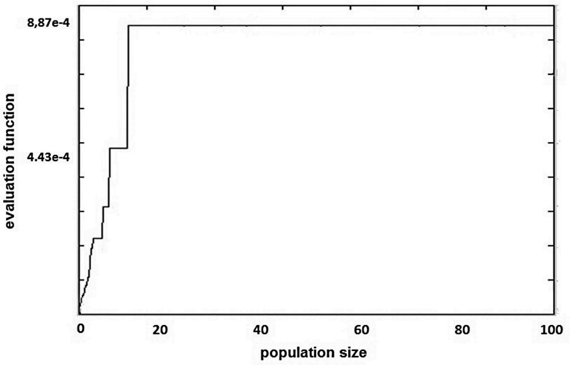

Fig. 7. Dependence between the size of the population and the value of the GA evaluation function.

Time and frequency analyses was made, during which a set of symptoms were acquired for the classifier training. Characteristic points are extracted from the part of the output signal that is transient, because of its susceptibility to a change during the parametric sweep. It can also provide more information on the system's behavior than the steady state analysis. The following parameters were used as characteristic points:

- Time domain analysis—the value of the first two maximal and minimal values in the output signal and the time instants required to reach these values (Fig. 8). Zero-crossing coordinates were also considered.

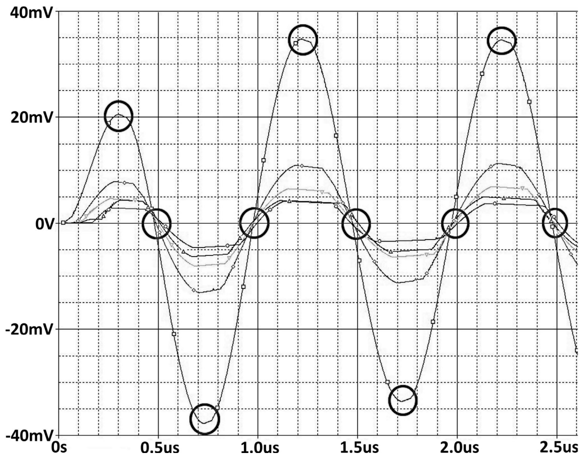

Fig. 8. Example shows how the output signal of a 52-element low-pass filter changes with the increase of the value of R2 in the range of 100–900 ohms. The characteristic points are also indicated.

- Frequency domain analysis—the 3 db frequency was measured.

The sine input signal was used as excitation. The proposed symptoms allow acquiring the most important information from the output signal. After changing the value of the selected parameter, the output of the SUT was recorded and the characteristic points extracted. These data were used for determination of the internal state of the SUT. The procedure was repeated for every element separately (the remaining elements were at nominal values).

The diagnostic method is divided into two parts. The first one, identical for any simulated system, is responsible for generating the data set and performing SVM training. There is no training phase for the $k$NN classifier. The created set provides learning data for the second part of the method during which classifiers are tested for their generalization abilities. The SVM toolbox used in this research was authored by Steve Gunn. Its kernel parameters were set according to Bilski (2013). In order to conduct the optimization of the SVM kernel parameter value, a quasi-discrete method was utilized. The range of kernel parameter values was divided into 100 intervals, and from each, one value was selected. The value providing the best classification results was returned as the optimization outcome.

## 4. Description of analyzed SUTs

The studies presented in this paper were conducted on a 26-element video amplifier (Fig. 9) and a 52-element low-pass filter (Fig. 10), both being modeled using the electronic schematics creation tool, Orcad PSpice 9.1. These linear objects were selected based on the increasing number of parameters they contain. The accessible nodes are visible in Figs. 9 and 10. The circuit of Fig. 9 includes 9 capacitors, 6 inductances, 9 resistors and 2 PNP bipolar junction transistors, with the amplification coefficient $k = 3$. They are hybrid-pi models. The circuit works in the range of 1 to 6 MHz and is excited by a sinusoidal voltage source of amplitude 10 mV. The feedback loop decreases the amplification, nonlinearities and distortions. This system has 13 accessible nodes.

The system in Fig. 10 is a 52-element fourth-order elliptic active low-pass filter with 19 nodes. It consists of 8 operational amplifiers modeled by a controlled source (VCCS) and the input and output resistances. The faults modeled in these amplifiers provide information about which VCCS is corrupted. The system was excited by a 10 mA current source of a 10 kHz frequency. Both circuits were modeled in PSpice.

## 5. Experimental results

The details of the experiment presented for both diagnosed systems are in Table 7. Both training and testing datasets represent knowledge collected from a single node. The optimal configuration of nodes is in Tables 8 and 9. This solution contains nodes that should be excited and measured to get better results than just the input-output analysis in the shortest time possible. The process of acquiring a solution by using node selectors (tabu search or the GA) was divided into two stages. In the first one, the node optimization process was conducted with omission of the output node. In the second stage, the output node was considered in the experiment. The aim of that was to examine how the ability of detecting a fault in the complex system increases with the inclusion of inner nodes in the analysis. Subsequently, fault classification efficiency was analyzed and compared with the previously conducted input-output only analysis (Tables 10–12). The best diagnostics results can be achieved by using the SVM with RBF kernel functions.

In the node selection process, some nodes can be replaced by others without any loss in the quality of fault detection. In the video amplifier system, node 7 can be replaced with node 8. The same can be said about nodes 8 and 11. The inclusion of nodes 9 or 10 does not increase diagnostic efficiency. The most crucial nodes for diagnostics are 2, 3 and 4. They are located in the part of the circuit where the input signal is strengthened by operational amplifiers. These nodes have a high degree; from them the change in the significant number of parameters can be observed. Node 12 plays an important role in the diagnostic process only when the parameter value of the element that is part of the feedback loop is disturbed.

In the case of the 52-element low-pass filter, nodes 2 and 3, 4 and 5, 6 and 7, 9 and 10, 13 and 14, 15 and 16, 18 and 19 can be excited interchangeably, because it is less significant for the diagnostic process. The inclusion of the output node in the solutions presented in Tables 2 and 3 increases the classification efficiency of the parametric fault in the low-pass filter system to 79% .

The best $k$NN classification results were acquired while using the Euclidean and Manhattan norms as means of calculating the distance between the examples. The categories were assigned to examples by an uneven number of neighbors to make the classification easier. The optimal number of neighbors $k$ was 5. The best classification efficiency achieved for the $k$NN algorithm was 70%. Because there is no training phase in the $k$NN algorithm, its performance is low. The classification quality achieved by SVM Gaussian kernels is much higher than in the case of $k$NN. Fault classification with the polynomial kernel requires polynomials of a higher degree (at least 33). This indicates a high level of complexity

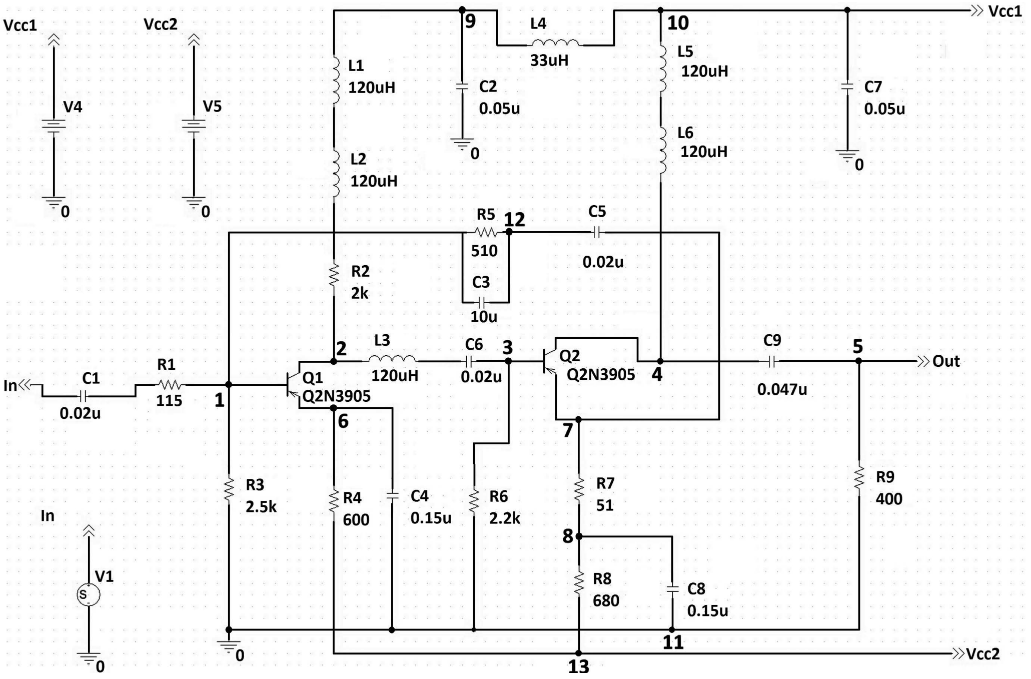

Fig. 9. 26-element video amplifier system with 13 accessible nodes modeled in Spice.

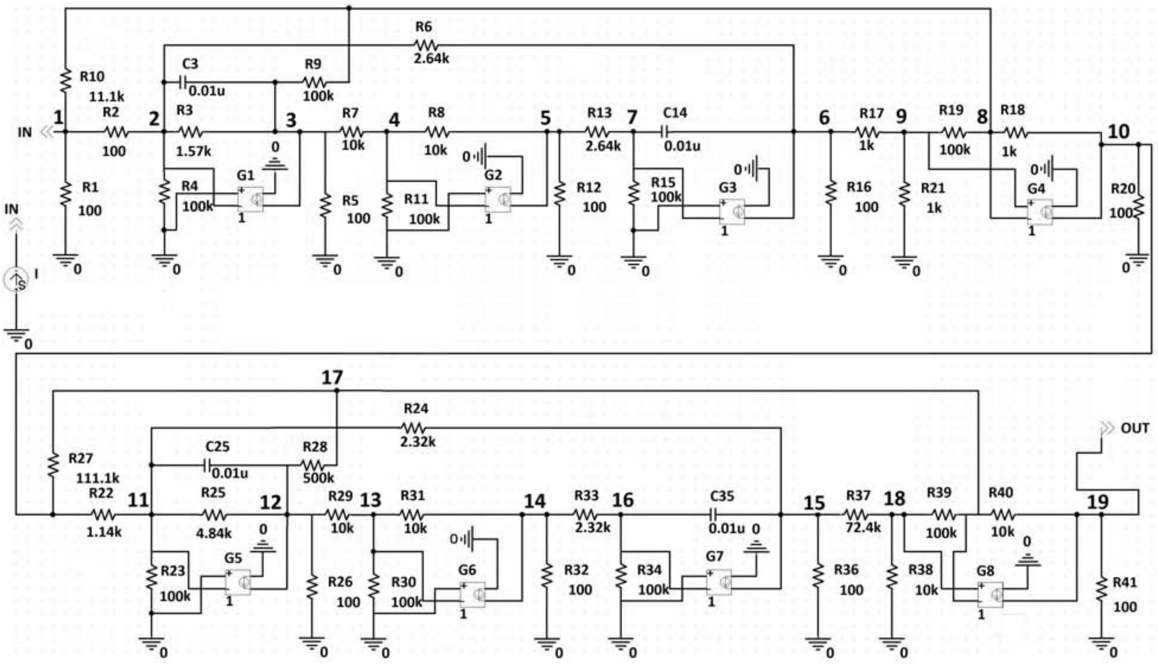

Fig. 10. 52-element low-pass active filter with 19 accessible nodes modeled in Spice.

Table 7. Details of the experiment presented for diagnosed systems.

| Size of the tested system | Size of the learning set | Size of the testing set |
|---|---|---|
| 26-element video amplifier | 176 | 88 |
| 52-element low-pass filter | 416 | 208 |

Table 8. Best node selection for a 26-element video amplifier, provided by the GA and tabu search simulation.

| Node number/node selector | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| GA | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| tabu | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |

Table 9. Best node selection for a 52-element low-pass filter, provided by the GA and tabu search simulation.

| Node number/node selector | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| GA | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| tabu | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |

Table 10. Classification efficiency for the input-output only analysis, provided by the SVM and $k$NN fault classifiers.

| Algorithm | Kernel parameter | Parameter value | Efficiency (percent) |
|---|---|---|---|
| 26-element video amplifier | | | |
| SVM (RBF) | width of RBFs | $0.000039 - 0.000096$ | 85 |
| SVM (ERBF) | width of RBFs | $0.00108 - 0.00207$ | 85 |
| SVM (poly) | degree of polynomial | $33 - 103$ | 60 |
| $k$NN (cityblock) | number of neighbors | 5 | 60 |
| $k$NN (Euclidean) | number of neighbors | 5 | 60 |
| 52-element low-pass filter | | | |
| SVM (RBF) | width of RBFs | $0.000039 - 0.000096$ | 71 |
| SVM (ERBF) | width of RBFs | $0.00108 - 0.00207$ | 71 |
| SVM (poly) | degree of polynomial | $33 - 103$ | 61 |
| $k$NN (cityblock) | number of neighbors | 5 | 60 |
| $k$NN (Euclidean) | number of neighbors | 5 | 60 |

Table 11. Classification efficiency for a 26-element video amplifier for optimal node configuration, provided by the SVM and kNN fault classifiers.

| Algorithm | Kernel parameter | Parameter value | Efficiency (percent) |
|---|---|---|---|
| output node excluded | | | |
| SVM (RBF) | width of RBFs | $0.000039 - 0.000096$ | 88 |
| SVM (ERBF) | width of RBFs | $0.00108 - 0.00207$ | 88 |
| SVM (poly) | degree of polynomial | $33 - 61$ | 70 |
| $k$NN (cityblock) | number of neighbors | 5 | 65 |
| $k$NN (Euclidean) | number of neighbors | 5 | 65 |
| output node included | | | |
| SVM (RBF) | width of RBFs | $0.000039 - 0.000096$ | 89 |
| SVM (ERBF) | width of RBFs | $0.00108 - 0.00207$ | 89 |
| SVM (poly) | degree of polynomial | $33 - 61$ | 71 |
| $k$NN (cityblock) | number of neighbors | 5 | 65 |
| $k$NN (Euclidean) | number of neighbors | 5 | 65 |

of data transformation from the original feature space so that linear nonseparability of the classified examples is retained. The complexity of the hyperplane separating different categories in the original feature space grows with example separation difficulty. Therefore, it can be assumed that the high order of the polynomial used in complex analog system diagnostic relates to the complexity of the analyzed system and requires more specialized functions. Smaller systems were successfully diagnosed using polynomial kernels of a lower order. The classification results for the RBF and ERBF kernels are the same, but require different parameters to make

Table 12. Classification efficiency for a 52-element low-pass filter for optimal node configuration, provided by the SVM and $k$NN fault classifiers.

| Algorithm | Kernel parameter | Parameter value | Efficiency (percent) |
|---|---|---|---|
| output node excluded | | | |
| SVM (RBF) | width of RBFs | 0.0000394−0.000096 | 78 |
| SVM (ERBF) | width of RBFs | 0.00108−0.00207 | 78 |
| SVM (poly) | degree of polynomial | 33−42 | 70 |
| kNN (cityblock) | number of neighbors | 5 | 65 |
| kNN (Euclidean) | number of neighbors | 5 | 65 |
| output node included | | | |
| SVM (RBF) | width of RBFs | 0.000039−0.000096 | 79 |
| SVM (ERBF) | width of RBFs | 0.00108−0.00207 | 79 |
| SVM (poly) | degree of polynomial | 33−42 | 71 |
| $k$NN (cityblock) | number of neighbors | 5 | 66 |
| $k$NN (Euclidean) | number of neighbors | 5 | 66 |

them optimal. For the Gaussian and polynomial kernels, a significant improvement in fault diagnostics can be observed when nodes of higher order (such as 3, 7 or 8) are included in the diagnostics.

Node selection using tabu search achieved similar results to the GA, but did it much quicker. Being a less computationally complex algorithm that the two used in the experiment, tabu seems the proper optimization algorithm for the task of node selection in complex analog systems.

## 6. Conclusions

This work shows a method to increase parametric fault classification efficiency in complex analog systems using minimum node selection compared to the input-output analysis. The minimal number of nodes required to improve the fault classification efficiency is 40% of all nodes in the case where the output node is omitted. In the presented systems, it was 5 out of 13 nodes for a video amplifier and 9 out of 19 nodes for low-pass filter. It is a result of time limitations that enforces the assumed evaluation function. During the simulation including the output node, the desirable fault classification efficiency could be achieved within around 23–26% of the total number of nodes.

The types of nodes to be included in the diagnostic process are those of a high degree, which allows the expert system to acquire information from a substantial number of elements connected to them. The method presented here can also be used in amplifying circuits with feedback loops. The increase in the number of nodes above the threshold considered crucial for achieving better diagnostic efficiency than the one provided by the input-output analysis only slightly improves the classification efficiency. The reason behind it is the fact that parametric faults are hard to detect.

The classification tool most efficient in complex analog systems diagnostic procedures is the SVM classifier equipped with Gaussian kernels. Fault classification results provided by this AI method demonstrate low generalization abilities, but that is to be expected for this type of data. It is a typical situation in complex analog systems diagnostics with parametric faults. The $k$NN algorithm, though easy to implement, lacks the ability to properly classify faults in measurement uncertainty conditions and thus it can be branded unsuitable for the purposes of complex analog systems diagnostics.

## Acknowledgment

The work of A. Bilski was supported by the Polish National Science Center, grant no. 2011/03/D/ST8/04309.

## References

Aminian, F. and Modular, A.(2007). Fault-diagnostic system for analog electronic circuit using neural networks with wavelet transform as a preprocessor, *IEEE Transactions on Instrumentation and Measurement* **56**(5): 1546–1554.

Arabas, J. (2004). *Lectures in Evolutionary Algorithms*, WNT, Warsaw, (in Polish).

Bilski, A. (2013). Diagnostic of complex analog systems with parametric faults using support vector machines, *in* T. Kwater and B. Twaróg (Eds.), *Computing in Science and Technology 2012/13*, University of Rzeszow, Rzeszów, pp. 7–24.

Bilski, P. (2007). Automated diagnostic system using graph clustering algorithm and fuzzy logic method, *18th European Conference on Circuit Theory and Design 2007, Seville, Spain*, pp. 779–782.

Bilski, P. (2011). Automated selection of kernel parameters in diagnostics of analog systems, *Przegląd Elektrotechniczny* **87**(5): 9–13.

Bilski, P. and Wojciechowski, J. (2007). Automated diagnostics of analog systems using fuzzy logic approach, *IEEE*

*Transactions on Instrumentation and Measurement* **56**(6): 2175–2185.

Bilski, P. and Wojciechowski, J. (2012). Current research trends in diagnostics of analog systems, *2012 International Conference on IEEE Signals and Electronic Systems (ICSES), Wrocław, Poland*, pp. 1–11.

Bushell, L. and Vishwani, D.A. (2002). *Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits*, Springer US, New York, NY.

Czaja, Z. and Zielonko, R. (2004). On fault diagnosis of analogue electronic circuits based on transformations in multidimensional spaces, *Measurement* **35**(3): 293–301.

Chakrabarti, S., Cherubal, S. and Chatterjee, A. (1999). Fault diagnosis for mixed-signal electronic systems, *IEEE Aerospace Conference, Snowmass at Aspen, CO, USA*, pp. 169–179.

Chatterjee, A., Kim, B. and Nagi, N. (1996 ). DC built-in self-test for linear analog circuits, *IEEE Design and Test of Computers* **13**(2): 26–33.

Fang, L., Plamen, K.N. and Sule, O. (2006 ). Parametric fault diagnosis for analog circuits using a Bayesian framework, *Proceedings of the 24th IEEE VLSI Test Symposium VTS'06, Berkeley, CA, USA,* pp. 272–277.

Gendreau, M. (2003). An introduction to tabu search, *in* F. Glover and G.A. Kochenberger (Eds.), *Handbook of Metaheuristics*, Springer, US, New York, NY, pp. 37–54.

Grzechca, D., Golonek, T. and Rutkowski, J. (2006). Analog fault AC dictionary creation—the fuzzy set approach, *IEEE International Symposium on Circuits and Systems, Kos, Greece*, pp. 5744–5747.

Grzechca, D., Golonek, T. and Rutkowski, J. (2007). Simulated annealing with fuzzy fitness function for test frequencies selection, *Proceedings of the IEEE Conference on Fuzzy Systems, London, UK*, pp. 1–6.

Grasso, F., Luchetta, A., Manetti, S. and Piccirilla, M.C. (2007). Method for the automatic selection of test frequencies in analog fault diagnosis, *IEEE Transactions on Instrumentation and Measurement* **56**(6): 2322–2329.

Golonek, T., Grzechca, D. and Rutkowski, J. (2008). Optimization of PWL analog testing excitation by means of genetic algorithm, *Proceedings of the International Conference on Signals and Electronic Systems, Kraków, Poland*, pp. 541–548

Golonek, T. and Rutkowski, J. (2007). Genetic-algorithm-based method for optimal analog test points selection, *IEEE Transactions on Circuits and Systems II* **54**(2): 117–121.

Guo, Y.-M., Wang, X.-T., Liu, Ch., Zheng, Y.-F. and Cai, X.-B. (2014). Electronic system fault diagnosis with optimized multi-kernel SVM by improved CPSO, *Maintenance and Reliability* **16**(1): 85–91.

Hochwald, W. and Bastian, J.D. (1979). A DC dictionary approach for analog fault dictionary determination, *IEEE Transactions on Circuits and Systems* **26**(7): 523–529.

Huertas, I.L (1993). Test and design for testability of analog and mixed-signal integrated circuits: Theoretical basis and pragmatical approaches, *Proceedings of the European Conference on Circuit Theory and Design, Davos, Switzerland*, pp. 1389–1407.

Huang, K., Stratigopoulos, H.-G. and Mir, S. (2010). Fault diagnosis of analog circuits based on machine learning, *2010 Design, Automation and Test in Europe Conference and Exhibition (DATE 2010), Dresden, Germany*, pp. 1761–1766.

Jantos, P., Grzechca, D. and Rutkowski, J. (2009). A global parametric faults diagnosis with the use of artificial neural networks, *European Conference on Circuit Theory and Design, Antalya, Turkey*, pp. 651–655.

Jantos, P., Grzechca, D. and Zielonko, R. (2009). Global parametric faults identification in analog electronic circuits, *Metrology and Measurement Systems* **16**(3): 391–402.

Korbicz, J., Obuchowicz, A. and Uciński, D. (1994). *Artificial Neural Networks. Fundamentals and Applications*, PLJ, Warsaw, (in Polish).

Kuczyński, A. and Ossowski, M. (2009). Analog circuits diagnosis using discrete wavelet transform of supply current, *Metrology and Measurement Systems* **16**(1): 77–85.

Milor, L.S. (1998). A tutorial introduction to research on analog and mixed-signal circuit testing, *IEEE Transactions on Circuits and Systems II* **41**(10): 1389–1407.

Nguyen, W.H. and Golinval, J.-C. (2010). Fault detection based on kernel principal component analysis, *Engineering Structures* **32**(11): pp. 3683–3691.

Osowski, S. (2006). *Artificial Neural Networks for Information Processing*, Warsaw University of Technology Press, Warsaw, (in Polish).

Ohletz, M. (1991). Hybrid built-in self-test for mixed analog/digital integrated circuits, *European Test Conference, Munich, Germany*, pp. 307–316.

Pan, C. and Cheng, K.-T. (1995). Pseudorandom testing and signature analysis for mixed-signal systems, *IEEE International Conference on Computer-Aided Design, San Jose, CA, USA*, pp. 102–107.

Prasad, V.C. and Babu, N.S.C. (2000). Selection of test nodes for analog fault diagnosis in dictionary approach, *IEEE Transactions on Instrumentation and Measurement* **49**(6): 1289–1297.

Rutkowski, J. and Grzechca, D. (2009). Fault diagnosis in analog electronic circuits—the SVM approach, *Metrology and Measurement Systems* **16**(4): 583–598.

Salama, A.E., Starzyk, J.A. and Bandler, J.W. (1984). A unified decomposition approach for fault location in large analog circuits, *IEEE Transactions on Circuits and Systems* **31**(7): 609–622.

Sałat, R. and Osowski, S. (2011). Support vector machine for soft fault location in electrical circuits, *Journal of Intelligent and Fuzzy Systems* **22**(1): 21–31.

Sen, N. and Saeks, R. (1979 ). Fault diagnosis for linear systems via multifrequency measurements, *IEEE Transactions on Circuits and Systems* **26**(7): 457–465.

Starzyk, J.A. and Dai, H. (1992 ). A decomposition approach for testing large analog networks, *Journal of Electronic Testing: Theory and Applications* **3**(3): 181–195.

Starzyk, J.A., Liu, D., Liu, Z.-H., Nelson, D.E. and Rutkowski, J. (2004). Entropy-based optimum test points selection for analog fault dictionary techniques, *IEEE Transactions on Instrumentation and Measurement* **53**(2): 754–761.

Sun, J., Wang, Ch., Sun, J. and Wang, L. (2013). Analog circuit soft fault diagnosis based on PCA and PSO-SVM, *Journal of Networks* **8**(12): 2791–2796.

Spina, R. and Upadhyaya, S. (1997). Linear circuit fault diagnosis using neuromorphic analyzers, *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing* **44**(3): 188–196.

Tadeusiewicz, M., Hałgas, S. and Korzybski, M. (2011). Multiple catastrophic fault diagnosis of analog circuits considering the component tolerances, *International Journal of Circuit Theory Application* **44**(3): 188–196.

Tadeusiewicz, M. and Korzybski, M. (2000). A method for fault diagnosis in linear electronic circuits, *International Journal of Circuit Theory and Applications* **28**(3): 245–262.

Tadeusiewicz, M. and Hałgas, S. (2006). An algorithm for multiple fault diagnosis in analogue circuits, *International Journal of Circuit Theory and Applications* **34**(6): 607–615.

Widodo, A. and Bo-Suk, T. (2007). Support vector machine in machine condition monitoring and fault diagnosis, *Mechanical Systems and Signal Processing* **21**(6): 2560–2574.

Wang, P. and Yang, S. (2005). A new diagnosis approach for handling tolerance in analog and mixed-signal circuits by using fuzzy math, *IEEE Transactions on Circuits and Systems I: Regular Papers* **52**(10): 2118–2127.

Vapnik, V. and Cortes, C. (1995). Support-vector networks, *Machine Learning* **20**(3): 273–297.

**Adrian Bilski** received his M.Sc. degree from the Faculty of Electrical Engineering, Warsaw University of Technology, Poland, in 2010. In 2016 he received his Ph.D. degree from the Faculty of Electronics at the same university. He currently holds the position of an assistant professor in the Warsaw University of Life Sciences, Faculty of Applied Mathematics and Informatics. His main scientific interests are diagnostics of analog systems as well as applications of artificial intelligence and machine learning methods to environmental sciences.

**Jacek Wojciechowski** (1942–2015) received his M.Sc., Ph.D. (with honors), and D.Sc. degrees from the Faculty of Electronics, Warsaw University of Technology, Poland, in 1966, 1976, and 1989, respectively. He also received an M.Sc. degree in mathematics from the University of Warsaw in 1975. He was a full professor at the Institute of Radioelectronics, Warsaw University of Technology. He was the author or a coauthor of over 100 technical publications and six books. His research interests covered systems engineering (simulation, and design for quality, sensitivity analysis, fault location in analog, and switched systems), information and communication systems, networks and graphs, and heuristic methods (e.g., evolutionary algorithms and fuzzy logic).

Received: 24 October 2014
Revised: 21 June 2015
Re-revised: 21 December 2015
Accepted: 9 March 2016