12th IMEKO TC1 & TC7 Joint Symposium on
Man Science & Measurement
September, 3 – 5, 2008, Annecy, France

A FAULT DIAGNOSIS ALGORITHM OF ANALOG CIRCUITS BASED ON NODE-VOLTAGE RELATION

Zbigniew Czaja

Gdansk University of Technology, Faculty of Electronics, Telecommunications and Informatics, Department of Optoelectronics and Electronic Systems, Gdansk, Poland, zbczaja@pg.gda.pl

Abstract: A new method of diagnosis of single faults of passive elements in analog electronic circuits, based on the node-voltage relation approach, is presented. This method consists of two parts: creation of a fault dictionary describing the nominal state of the tested circuit and containing indirect parameters representing respective faults, and a new fault detection and localization algorithm.

Keywords: fault diagnosis, analog circuits

1. INTRODUCTION

The level of fault diagnosis of analog circuits and also analog parts and components of mixed-signal electronic systems is still insufficient in spite of continuous development in this field, especially for circuits with limited measurement accessibility. It follows from the fact that components of analog circuits have continuous values varying within [0, ∞), where values 0 and ∞ representing catastrophic faults can additionally change the topology of the circuit; stimulation and response signals are also continuous and they can assume the shape of any function. Additional problems of fault diagnosis are the presence of component tolerances and circuit nonlinearities.

At present, computer computation based on the Monte Carlo method [1,2] or an approach based on fuzzy logic [3,4] are more and more often used to solve the first problem. The second problem is resolved by the piecewise linearization approach [5,6]. In this case a nonlinear circuit is "converted" into a set of linear circuits.

However, for electronic linear circuits the relations between measured parameters and component parameters usually are nonlinear. So, the calculations are complex even for simple linear circuits, and their complexity grows faster than the number of circuit components [7]. For this reason SBT (simulation before test) methods usually are used, what cuts the testing time. Test procedures do not need big computing power because the considerable majority of calculations are made in the pre-testing state. The SBT methods base on an analytical approach [8,9], neural networks [10,11], genetic algorithms [12] and the fuzzy logic approach [4,13]. For analytical methods the localization bases on the determination of component values [9] or bases on comparison of indirect parameters [14].

One of the analytical SBT methods, which uses indirect parameters for localization, is the approach based on a node-voltage relation [14]. Its main advantage is the linearity dependence between measurement parameters of the circuit (its node voltages) and indirect parameters which describe relations between the node voltages. Additionally, one indirect parameter is attributed to the failure of one component. Thus, the number of these parameters is equal to the number of circuit components. Due to this, the fault dictionary is small and calculations made during the testing procedure are not complicated. Hence, this method can be used for self-testing of analog parts of electronic embedded systems, where control units cannot appoint great computing power and also data and program memory spaces for diagnosis procedures and fault dictionaries.

Thus, in the paper a new algorithm of the fault diagnosis of analog circuits based on this approach will be presented. It can be implemented in electronic embedded systems controlled by microcontrollers, digital signal processors (DSP) or system-on-chips (SoC).

2. PRINCIPLES OF THE NODE-VOLTAGE RELATION APPROACH

Let the analog circuit under test consist of p_1, .., p_i, .., p_I passive elements (components), where I is the number of circuit elements. The circuit is stimulated by a DC or AC stimulus and voltages v_A and v_B of two nodes A and B are measured. The relations, as described in [14], between v_A and p_i and between v_B and p_i can be written as:

$$\begin{cases} v_{Ai} = f_{Ai}(p_i) \\ v_{Bi} = f_{Bi}(p_i) \end{cases} \tag{1}$$

If the inverse function for any of the two functions in (1) exists, what was fulfilled for linear analog circuits, we can eliminate the variable p_i from (1). For this purpose we can inverse e.g. the first function $p_i = f_{Ai}^{-1}(v_{Ai})$ and put in to the second line of (1):

$$v_{Bi} = f_{Bi}(f_{Ai}^{-1}(v_{Ai})) \tag{2}$$

The formula (2) is called the V^2 relation function [14]. This function does not contain the parameter p_i. So, it is determined by parameters of the remaining elements

$\{p_j\}_{j=1,..,I \text{ and } j\neq i}$ and the topology of the circuit $f_{Bi}(f_{Ai}^{-1}(\cdot))$. This is called the invariance of the V^2 relation function.

It is well known that the voltage transfer function (describes the relation between voltages in the input node and the output node [15] and also between voltages in the input node and internal nodes [16]) of a linear electronic circuit can be described in the form of a bilinear function of the value of each circuit component parameter p_i:

$$f_i(p_i,s)=\frac{v_{Xi}}{v_{in}}=\frac{\alpha_i(s)p_i+\beta_i(s)}{\chi_i(s)p_i+\delta_i(s)} \tag{3}$$

where: α_i, β_i, χ_i, δ_i are complex coefficients, $\alpha_i\delta_i$-$\beta_i\chi_i\neq 0$, v_{in} – input voltage, v_{Xi} – voltage in the X node, $X=\{A, B, C, \ldots\}$.

Thus, the voltage in the X node can be described as:

$$v_{Xi}=\frac{\alpha_i p_i+\beta_i}{\chi_i p_i+\delta_i}\cdot v_{in}=\frac{\alpha_i v_{in}p_i+\beta_i v_{in}}{\chi_i p_i+\delta_i}=\frac{\frac{\alpha_i v_{in}}{\chi_i}p_i+\frac{\beta_i v_{in}}{\chi_i}}{p_i+\frac{\delta_i}{\chi_i}} \tag{4}$$

Making the assumption that $a_{Xi}=\frac{\alpha_i v_{in}}{\chi_i}, b_{Xi}=\frac{\beta_i v_{in}}{\chi_i}, d_{Xi}=\frac{\delta_i}{\chi_i}$ we obtain the relation:

$$v_{Xi}=f_{Xi}(p_i)=\frac{a_{Xi}p_i+b_{Xi}}{p_i+d_{Xi}} \tag{5}$$

Hence, for linear analog circuits the relation between the node voltage v_{Xii} and the parameter p_i (5) is a hyperbolic function, where the voltage takes real values for purely resistive circuits under a DC stimulus, and for linear RC circuits under an AC stimulus it is a complex number.

For two nodes A and B we can write:

$$\begin{cases} v_{Ai}=f_{Ai}(p_i)=\frac{a_{Ai}\cdot p_i+b_{Ai}}{p_i+d_{Ai}} \\ v_{Bi}=f_{Bi}(p_i)=\frac{a_{Bi}\cdot p_i+b_{Bi}}{p_i+d_{Bi}} \end{cases} \tag{6}$$

and functions $f_{Ai}(\cdot)$ and $f_{Bi}(\cdot)$ have the same denominator [14] $d_{Ai}=d_{Bi}$. Proceeding accordingly with the formula (2), from the first function of (6), we determine the p_i parameter:

$$p_i=\frac{b_{Ai}-d_{Ai}\cdot v_{Ai}}{v_{Ai}-a_{Ai}} \tag{7}$$

Substituting (7) into the second function of (6), we get the V^2 relation function:

$$v_{Ai}=\frac{a_{Ai}\cdot d_{Bi}-b_{Ai}}{a_{Bi}\cdot d_{Ai}-b_{Bi}}v_{Bi}+\frac{a_{Bi}\cdot b_{Ai}-a_{Ai}\cdot b_{Bi}}{a_{Bi}\cdot d_{Ai}-b_{Bi}} \tag{8}$$

The relation function (8) is a linear function $v_{Ai}=k_i\cdot v_{Bi}+b_i$ on the v_1-v_2 plane, where the slope k_i and an intercept b_i have the forms:

$$k_i=\frac{a_{Ai}\cdot d_{Bi}-b_{Ai}}{a_{Bi}\cdot d_{Ai}-b_{Bi}} \text{ and } b_i=\frac{a_{Bi}\cdot b_{Ai}-a_{Ai}\cdot b_{Bi}}{a_{Bi}\cdot d_{Ai}-b_{Bi}} \tag{9}$$

From relations (9) it is seen that parameters k_i and b_i do not depend on the p_i parameter, they are only related to other components' parameters and the circuit topology.

Assuming that a fault set has the form $\boldsymbol{F}_I=\{F_i\}_{i=0,1,..,I}$, where for $i=0$ we have a fault-free state (a nominal state) and F_i denotes a failure of the p_i element (a catastrophic fault or a soft fault (parametric deviation)). When the circuit is in the nominal state, voltages in nodes A and B have the following values v_{Anom} and v_{Bnom} respectively. For respective faults in the fault set we obtain different V^2 relation functions. Their graphs cross the same point (v_{Anom}, v_{Bnom}) named the nominal point in the v_1-v_2 plane. Because they are linear functions, we can write them in the form:

$$v_{Anom}-v_{Ai}=k_i\cdot(v_{Bnom}-v_{Bi}), \text{ where } i=1,..,I \tag{10}$$

Therefore, the slope k_i can be assigned from the relation:

$$k_i=\frac{v_{Anom}-v_{Ai}}{v_{Bnom}-v_{Bi}}, \text{ where for all } i\text{: } v_{Bnom}\neq v_{Bi} \tag{11}$$

Hence, the slope k_i can be used as a fault character (indirect parameter) for the F_i fault state, that is k_i is related to the fault of the i-th element. The set $K_I=\{k_i\}_{i=0,1,..,I}$ can be determined in the pre-testing stage and treated as a fault dictionary. So, generally the fault diagnosis of a single fault case bases on measurements of values v_{Ameas}, v_{Bmeas} of voltages in nodes A and B, and an assignment of the k_{meas} parameter bases on (11), which is compared with the set K_I. The near value k_i ($k_{meas}\approx k_i$) shows the fault of the i-th element.

As mentioned above for a linear circuit, node voltages are complex numbers. Hence, the slope k_i is also a complex number. Thus, it can be written: $k_i=\kappa_i+j\cdot\gamma_i$, where: $\kappa_i=\mathrm{Re}(k_i)$, $\cdot\gamma_i=\mathrm{Im}(k_i)$.

3. FAULT DIAGNOSIS PROCEDURES

The diagnosis method of single faults based on the node-voltage relation consists of two stages:

- Pre-testing stage, where the fault dictionary of the tested analog circuit is created.
- Fault detection and fault localization stages basing on measurement results and the fault dictionary, where first the fault detection is made, and when a fault is detected, the localization of a single fault is carried out.

The second stage was divided into two parts, because in this method to localize a fault we have to know that the circuit is faulty, what is explained below.

We know that the parameter k_i depends on all circuit parameters without the parameter p_i. If k_{meas} is calculated from measurement results and if $k_{meas}\approx k_i$, it means only that all elements of the tested circuit without the p_i element have nominal values. Hence, the p_i element can, but does not have to be faulty (k_i is not a function of p_i). But if we know that the circuit is faulty (after a detection procedure) and $k_{meas}\approx k_i$, then the p_i element is faulty, because the remaining elements are not faulty.

3.1. Creation of the fault dictionary

The method will be illustrated on the analog circuit shown in Fig. 1. The low-pass active filter is stimulated by a sinusoidal signal with a frequency $f=735$ Hz and an amplitude $V_{in}=1$ V (Fig. 2). Nodes A and B are marked in Fig. 1, where only node A is an internal node. This solution simplifies the design of the circuit and measurement

procedures, what is especially important for electronic embedded systems.

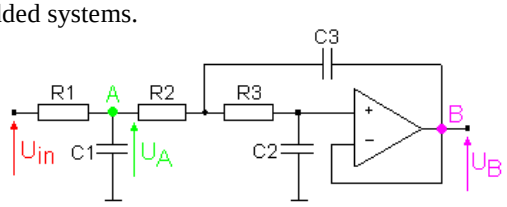


Fig. 1. The tested analog circuit, where R1=R2=R3=5 kΩ, C1=44.5 nF, C2=6.42 nF, C3=110 nF

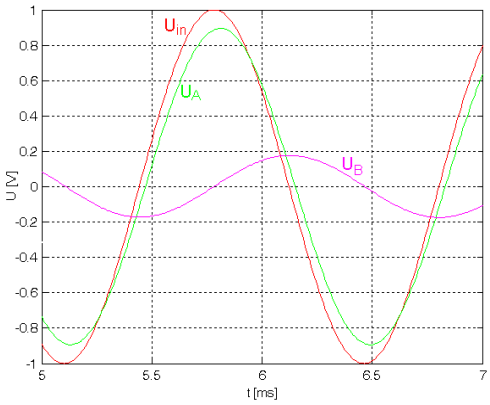


Fig. 2. Timings in nodes A and B for the nominal state of the circuit from Fig. 1.

In the first step of creation of the fault dictionary we generate data used by the detection algorithm, that is, the description of the circuit in the nominal state. Fig. 3 illustrates nominal areas which represent the nominal state in nodes A and B. They were drawn based on the Monte Carlo method with regard to 1% tolerances of resistors and capacitors.

Thus, if voltages measured in nodes A and B, that is, the measurement point P_{Ameas} in node A with coordinates $(\mathrm{Re}(v_{Ameas}), \mathrm{Im}(v_{Ameas}))$ and the measurement point P_{Bmeas} $(\mathrm{Re}(v_{Bmeas}), \mathrm{Im}(v_{Bmeas}))$ in the node B, both are placed inside nominal areas of respective nodes, it means that the tested circuit is fault-free.

Hence, data representing the circuit in the nominal state should describe nominal areas for both nodes. It was assumed that each nominal area is described by an approximation circle with a centre P_{Xnom} and a radius R_X, where X signifies any node (Fig. 3).

Determination of parameters of the approximation circle is realized by the following algorithm:

- Coordinates $(\mathrm{Re}(v_{Xnom}), \mathrm{Im}(v_{Xnom}))$ of the centre P_{Xnom} are calculated for the nominal state of the circuit.
- M points with coordinates $(\mathrm{Re}(v_{Xm}), \mathrm{Im}(v_{Xm}))$, m=1, .., M, representing the nominal area are generated using the Monte Carlo method.
- The length of the radius R_X is determined from the relationship:

$$R_X = \max\{(\max_{m=1,..,M}\{\mathrm{Re}(v_{Xm})\} - \min_{m=1,..,M}\{\mathrm{Re}(v_{Xm})\}), (\max_{m=1,..,M}\{\mathrm{Im}(v_{Xm})\} - \min_{m=1,..,M}\{\mathrm{Im}(v_{Xm})\})\}/2 \quad (12a)$$

or

$$R_X = \min\{(\max_{m=1,..,M}\{\mathrm{Re}(v_{Xm})\} - \min_{m=1,..,M}\{\mathrm{Re}(v_{Xm})\}), (\max_{m=1,..,M}\{\mathrm{Im}(v_{Xm})\} - \min_{m=1,..,M}\{\mathrm{Im}(v_{Xm})\})\}/2 \quad (12b)$$

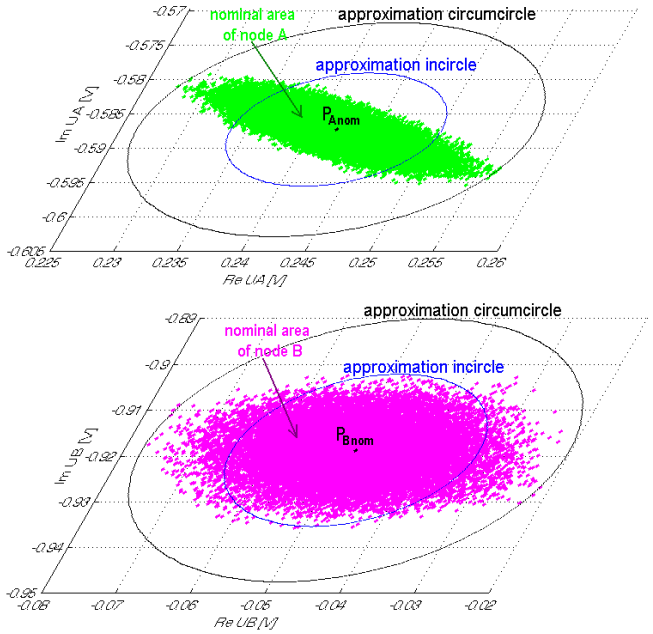


Fig. 3. Nominal areas of the node A and the node B for the tested analog circuit with approximation circles

In the case (12a) the circle is circumscribed on the nominal area. Certainly, we can change the external operator "max" to "min", what implies that the circle will be inscribed into the nominal area (12b). The choice between the approximation circumcircle and incircle can depend on application requirements. For instance, if the embedded system is used in critical applications (e.g. in medicine, in aerospace) we should decide to use the incircle, because for these applications reliability is the most important, and when the circuit still works correctly but on a boundary of elements tolerances, it is better to treat it as faulty than to allow conversion errors. For custom applications we can use the first case to eliminate unnecessary alarms, especially when the circuit works correctly.

At the end of this step of creation of the fault dictionary, the following parameters of approximation circles are written to the fault dictionary: $\{v_{Anom}, v_{Bnom}, R_A, R_B\}$, where v_{Anom}, v_{Bnom} are complex numbers. If we write that $v_{Xnom} = u_{Xnom} + j \cdot w_{Xnom}$, the fault dictionary has the form: $\{u_{Anom}, w_{Anom}, u_{Bnom}, w_{Bnom}, R_A, R_B\}$.

Generation of the slope set K_I (localization coefficients) is realized in the second step of creation of the fault dictionary. Calculations of k_i base on the modified relation (11):

$$k_i = \frac{v_B(p_{i\,nom}) - v_B(\xi \cdot p_{i\,nom})}{v_A(p_{i\,nom}) - v_A(\xi \cdot p_{i\,nom})} \quad (13)$$

where $v_A(\cdot)$, $v_B(\cdot)$ are calculated voltage values for given values of the p_i element in nodes A and B, p_{inom} – the nominal value of the p_i element, ξ – the assumed deviation of the p_i element value. In the paper it was assumed that ξ is equal to 10.

Based on this relation (13) I localization coefficients are calculated. Thus, after this step we add to the fault dictionary the set $\{k_i\}_{i=0,1,..,I}$ of complex coefficients, where we write that $k_i = \kappa_i + j \cdot \gamma_i$.

Hence, finally the full fault dictionary has the following form: $\{u_{Anom}, w_{Anom}, u_{Bnom}, w_{Bnom}, R_A, R_B, \{\kappa_i, \gamma_i\}_{i=0,1,..,I}\}$. It will be used by the fault detection and localization algorithm.

3.2. The fault detection and localization algorithm

Fault detection and fault localization stages are realized by the algorithm shown in Fig.4.

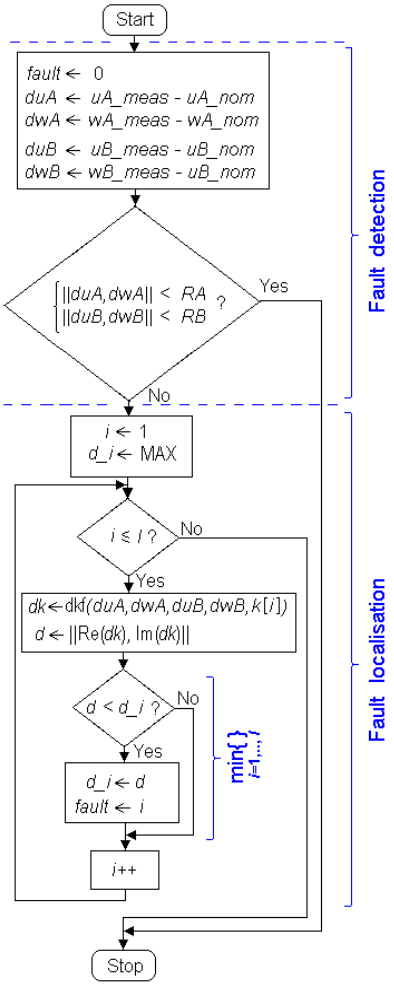


Fig. 4. Detection and localization algorithm of a single soft fault

It is seen (Fig. 4) that the algorithm consists of two parts. The first part is responsible for the fault detection. If it detects a fault, the second part of the algorithm is running and the fault is localized.

At the beginning of the fault detection part the variable *fault* containing the number (the index) of the faulty element is cleared. Next, differences kept by variables *duA*, *dwA*, *duB*, *dwB* between measurement results (variables: *uA_meas*, *wA_meas*, *uB_meas*, *wB_meas*) in nodes *A* and *B* and nominal voltage values (variables: *uA_nom*, *wA_nom*, *uB_nom*, *wB_nom*) in these nodes are calculated. At the end of this part it is tested that the measurement result is placed inside the approximation circles, i.e.: is the result satisfying the following condition:

$$\begin{cases} \|du_A, dw_A\| < R_A \\ \|du_B, dw_B\| < R_B \end{cases} \tag{14}$$

where $\|\cdot\|$ is a norm defining the relation on a distance calculation. The taxi norm was assumed, because the described testing method is elaborated to implement it in 8-bit microcontrollers which do not have big computing power. It considerably simplifies calculations, because we use only three integer operations: subtraction, an absolute value and addition (represented by single microcontroller assembler instructions).

If condition (14) is satisfied, the algorithm is finished with the cleared variable *fault*, which means that the tested circuit is fault-free.

Else the fault localization part is run. At the beginning of the fragment of the localization algorithm the temporary variable *d_i* and the index variable *i* are initialized, where MAX is the maximal value for a given numerical system. It is assumed that variables are of the integer type (16-bit words), thus MAX = 7FFFh. Next, the following operations are repeated *I* times:

- The *dkf* function realizes the relationship (18b) and returns the result to the complex variable *dk*. From the localization condition $k_{meas} \approx k_i$ and (13) we know that:

$$k_i \approx k_{meas} = \frac{v_{Bmeas} - v_{Bnom}}{v_{Ameas} - v_{Anom}} = \frac{dv_B}{dv_A} \tag{15}$$

Thus, we can write:

$$k_i = k_{meas} + dk_i = \frac{dv_B}{dv_A} \tag{16}$$

Transforming (16) we obtain:

$$k_i - dk_i = \frac{dv_B}{dv_A} \tag{17}$$

what finally gives:

$$dk_i = dv_B - k_i \cdot dv_A \tag{18a}$$

We can write (18a) in the form:

$$dk_i = (du_B - \kappa_i \cdot du_A + \gamma_i \cdot dw_A) + j \cdot (dw_B - \kappa_i \cdot dw_A - \gamma_i \cdot du_A) \tag{18b}$$

Now the localization condition has the form $dk_i \approx 0$.

In this way we reduced the division operation and simplify testing of localization conditions. Hence, the *dkf* function bases on only addition, subtraction and multiplication operations, which can be directly implemented by assembler integer instructions of the microprocessors. This advantage allows to apply this approach in systems controlled by control units also with no big computer power (e.g. 8-bit microcontrollers).

- In the next step the normalized complex variable *dk* is kept by the variable *d*.
- At the end of the algorithm the *min*{} function is realized and the index of the element for which the variable *d* has a minimum value is written to the variable *fault*. Thence, the variable *fault* keeps the index of the faulty element. A comparison operation and determination of the minimum value are run currently, thus the *min*{} function needs only two variables, which saves the data memory space.

Thus, this algorithm is simple, that is, its code occupies a small place e.g. in the microcontroller program memory and

it bases on more than ten integer variables placed in the microcontroller RAM.

4. AN APPLICATION EXAMPLE

An application of the method for self-testing of the analog part of the embedded electronic system will be presented on an exemplary system controlled by the 8-bit microcontroller Atmega16 (Fig. 5).

This microcontroller has [17]: two advanced 8-bit timers/counters, one 16-bit timer/counter enabling precise measurements of time, the 10-bit SAR-type ADC with a sample & hold circuit (which gives the possibility of measurement of instantaneous voltage values) and with the 8-channel analog multiplexer, the analog comparator which can trigger the 16-bit timer/counter.

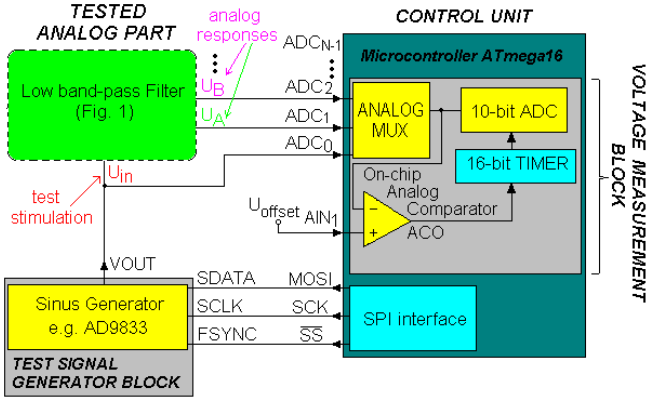


Fig. 5. Exemplary embedded electronic system controlled by the Atmega16 microcontroller

4.1. The measurement procedure

In the first step of the self-testing procedure the microcontroller controls measurements of node voltages. The measurements base on internal resources of the microcontroller creating the measurement microsystem (BIST) existing only during the testing time (the voltage measurement block in Fig. 5). The testing signal generator block (Fig. 5) controlled via the SPI interface generates a sinus wave with a frequency $f = 735$ Hz, an amplitude $U_{in} = 1$ V, and an offset $U_{offset} = 1$ V. The ADC connected to the appropriate node via the analog multiplexer samples voltages in moments established by the 16-bit timer/counter.

Many approaches are known how to determine sinus wave parameters, based on DFT, FFT and other digital sampling algorithms, e.g. [18]. Because the measurement results are used only for detection and localization of a faulty element, it is proposed to use a less accurate approach [19], but computing considerably simpler and faster. The paper presents only the idea of measurements of sine-wave parameters, because details of measurement algorithms, their implementation and experimental verification of the approach are described in [19].

The BIST controlled by the microcontroller generates the sine-wave stimulating the tested analog part (signal U_{in} in Fig. 6) and measures the amplitudes of voltage responses in nodes A and B (U_A, U_B respectively) and their time delays τ_A, τ_B in relation to the input signal U_{in}.

For each node the measurement procedure consists of the following steps (Fig. 6):

- The analog multiplexer connects the X node to the ADC input and to the analog comparator input.
- Detection of the u_X value above U_{offset} by the analog comparator starts the 16-bit timer.
- If the 16-bit timer counts off $T/4$, it triggers the ADC. $T = 1/f$.
- The ADC sample result has the value $U_{sample} = U_X + U_{offset}$. Thus the microcontroller calculates the voltage amplitude $U_X = U_{sample} - U_{offset}$.
- The analog multiplexer connects the input signal u_{in} to the analog comparator input.
- Detection of the u_X value above U_{offset} by the analog comparator starts the 16-bit timer to count the time delay τ_X.
- The analog multiplexer again connects the X node to the analog comparator input.
- Detection of the u_X value above U_{offset} by the analog comparator stops the 16-bit timer. Thus, the timer contains the value corresponding to the time delay τ_X.

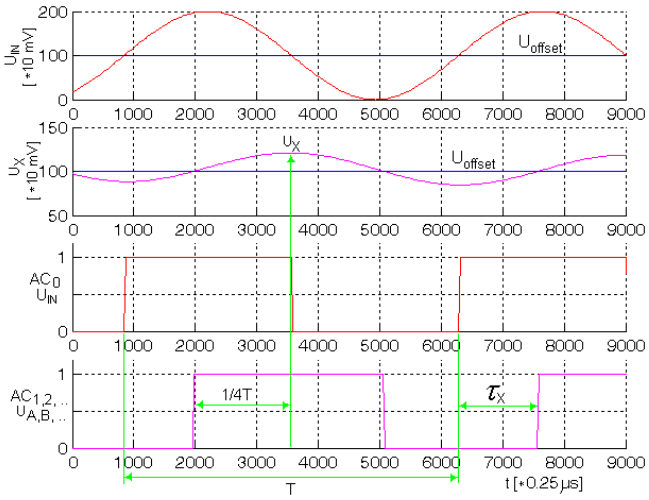


Fig. 6. Timings of the measurement procedure of voltage responses parameters in nodes *A* and *B*

Next, the amplitude U_X and the time delay τ_X are converted to real v_{Xmeas} and imaginary w_{Xmeas} values of the voltage in X node and restored in the form of integer 16-bit values.

4.2. Generation of the fault dictionary

Because the described fault diagnosis method belongs to SBT methods, the fault dictionary is created during the design of the embedded system and only once for a given analog circuit. It is generated on a PC computer using the Matlab program with the Control Toolbox, and is saved to a text file based on the *dlmwrite* function. Next, it is converted and placed into the file with the full code of a program.

For the tested circuit from Fig. 1, detection coefficients describing the approximation circumcircle $\{v_{Anom}, v_{Bnom}, R_A, R_B\}$ and the set $\{k_i\}_{i=0,1,..,I}$ of localization coefficients are shown in Table 1 and Table 2 respectively.

Table 1. Set of detection coefficients

	v_{Xnom}		R_X [V]
	u_{Xnom} [V]	w_{Xnom} [V]	
Node *A*	0.24194	-0.5873	0.0142
Node *B*	0.04885	-0.9187	0.0270

Table 2. Set $\{k_i\}_{i=0,1,..,I}$ of localization coefficients

	k_1 (R1)	k_2 (R2)	k_3 (R3)	k_4 (C1)	k_5 (C2)	k_6 (C3)
κ_i	1.3081	1.3692	2.5536	1.3081	2.4845	2.1060
γ_i	-0.6221	3.2680	0.6731	-0.6221	0.8246	0.7153

The Atmega16 does not support floating point calculations. Thus, it is proposed to convert floating point coefficients to integer values. This solution considerably reduces calculations and the procedure code size. Obviously, this conversion introduces rounding errors, but, as mentioned earlier, we use measurement results and the fault dictionary only at the level of fault detection and localization. Thus a 16-bit accuracy is completely sufficient.

It was assumed that coefficients u_{Xnom}, w_{Xnom}, R_X are multiplied by 32768, and the set $\{k_i\}_{i=0,1,..,I}$ by 128 and next rounded to integer values. This conversion influences only the calculation made by the *dkf* function (Fig. 4), which was considered in its code (Listing 5). The final version of the fault dictionary is presented in Listing 1.

```
// Detection coefficients
const int uA_nom = 7928,  wA_nom = -19245;
const int uB_nom = -1601, wB_nom = -30106;

const int RA = 465, RB = 886;

// Localization coefficients
#define I 6
const int re_k[I] = {167,175,327,167,318,270};
const int im_k[I] = {-80,418, 86,-80,106, 92};
```

Listing 1. Code of the fault dictionary written in ANSI C

It is seen that the fault dictionary consists of 18 constant integer values, which gives only 36 bytes in the program memory. It is very small in comparison with the size of the microcontroller program memory (16KB), which is the main advantage of the proposed diagnosis method.

4.2. Fault detection and localization procedures

After the measurement procedure the microcontroller runs firstly the detection procedure and next, if a fault is detected, the localization procedure. These procedures base on measurement results included in variables shown in Listing 2 and the fault dictionary (Listing 1) and they work according to the algorithm shown in Fig. 4.

The detection procedure is realized by the *detection* function shown in Listing 2.

```
#define MAX 0x0FF

typedef unsigned char uint8;

int duA, dwA, duB, dwB;
int uA_meas, wA_meas, uB_meas, wB_meas;
uint8 fault, selftest;

uint8 detection(void)
{
int rA, rB;

duA = uA_meas - uA_nom;
dwA = wA_meas - wA_nom;
duB = uB_meas - uB_nom;
dwB = wB_meas - wB_nom;

rA = distance(duA,dwA);
rB = distance(duB,dwB);

if((rA < RA) && (rB < RB)) return(0);
else return(MAX);
}
```

Listing 2. Code of the fault detection procedure written in ANSI C

This function exactly implements the first part of the algorithm shown in Fig. 4. It calculates subtractions between coordinates of the measurement point and the nominal point. Next, basing on the *distance* function it calculates the distances between these points for node *A* and node *B*. At the end, the function checks the position of the measurement point basing on condition (14). If this point is inside the approximation circle (condition (14) is fulfilled) it returns 0, else it returns MAX = FFh.

Thanks to the assumption of the taxi norm, the *distance* function is very simple. It uses only an absolute value and an integer addition operator, and also returns an integer value (Listing 3).

```
int distance(unsigned int x,unsigned int y)
{
x = fabs(x);
y = fabs(y);
return((x+y));    // taxi norm
}
```

Listing 3. Code of the *distance* function written in ANSI C

The second part "Fault localization" of the algorithm on Fig. 4 was written as the *localization* function (Listing 4).

```
#define MAXMAX 0x07FFF

uint8 localization(void)
{
uint8 i, fault_i;
int d_i, dk;

d_i = MAXMAX;
fault_i = MAX;

for(i=0;i<I;i++)
    {
       dk = dkf(re_k[i],im_k[i]);
    if(dk < d_i)
          {
          d_i = dk;
          fault_i = i + 1;
          }
       }
return(fault_i);
}
```

Listing 4. Code of the *localization* function written in ANSI C

This function initializes temporary variables *d_i* and *fault_i* with maximum values. Next, differences *dk* between slopes k_{meas} and k_i are calculated *I* times by the *dkf* function

in the loop “for”. In this loop the minimum value of the *dk* variable and corresponding to it the number (the index) of the circuit element are also determined. At the end, the function returns the number of the faulty element.

The *dkf* function realizes the relationship (18b) and additionally it returns the *dk* variable in the form of an absolute value (Listing 5).

```
#define N 7

int dkf(int re_ki,int im_ki)
{
int re_dk, im_dk, tuB, twB;

tuB = duB << N;     // 2 power N
twB = dwB << N;     // 2 power N

re_dk = tuB - re_ki*duA + im_ki*dwA;
im_dk = twB - re_ki*dwA - im_ki*duA;

return(distance(re_dk,im_dk));
}
```

Listing 5. Code of the *dkf* function written in ANSI C

The last listing (Listing 6) presents the exemplary code of the fragment with the code of the self-testing procedure of the main program controlling the embedded system.

```
/* main program */

int main(void)
{
    avr_init();       // initialization and
                      // configuration
                      // of the microcontroller
    selftest = 1;
      . . .
    while(1)
    {
      . . .
       if(selftest)
          {
          selftest = 0;  // clear selftest flag
          measuremet();  // measurement procedure
                         // described in [19]

          fault = detection();
          if(fault) fault = localisation();
          }
      . . .
    }
      . . .
    return(0);
}
```

Listing 6. Exemplary code of the fragment of the main program controlling the embedded system

It is seen that we propose to run the self-testing procedure after reset of the microcontroller. During the initialization part the *selftest* flag is set, what activates the self-testing procedure. This procedure clears the *selftest* flag and makes measurements (*measurement* function), next it detects the fault (*detection* function), and if the fault is detected, it runs the *localization* function.

After the self-testing procedure, the localization result can be e.g. displayed on any display provided in the system or it can be transmitted to a personal computer via any serial interface.

5. SIMULATION RESULTS

The effectiveness and robustness against element tolerances of the fault diagnosis algorithm based on the node-voltage relation was tested in a simulation way on the circuit from Fig. 1.

It was assumed that resistors and capacitors have 1% tolerances. Testing sets of measurement slopes $K_{meas} = \{k_{meas}{}^{m}\}_{m=1,..,M}$ for all circuit elements were generated using the Monte Carlo method (Fig. 7).

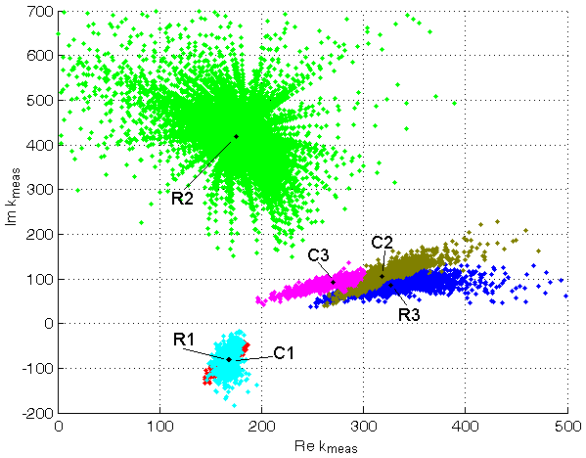
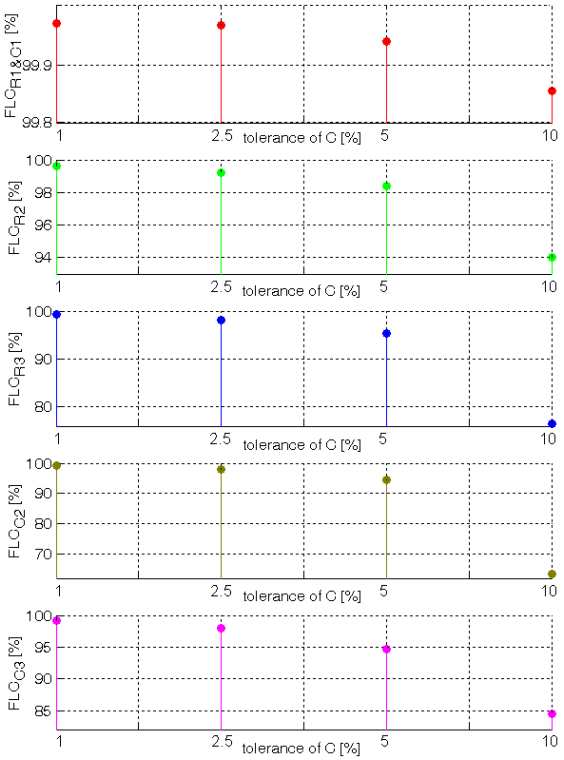


Fig. 7. Graphical illustration of testing sets of measurement slopes $\{k_{meas}{}^{m}\}_{m=1,..,M}$ for all circuit elements for the circuit from Fig. 1 for 1% tolerances of resistors and capacitors

From Fig. 7 it is seen that measurement slope sets for R1 and C1 elements overlap. It follows from the fact that these elements are not situated between nodes A and B and they cannot be distinguished from these nodes as separate elements (Fig. 1).



Listing 8. Simulation results of FLC_i coefficients for the circuit from Fig. 1 for 1% tolerance of resistors

The Fault Localization Covering coefficient of the *i*-th element (FLC_i) was introduced, defined as:

$$FLC_i[\%] = \frac{m_i}{M} \cdot 100\% \tag{19}$$

where: m_i – the number of corrected localized measurement points (measurement slopes) for the *i*-th element, M – the number of all generated points (measurement slopes) for the *i*-th element introduced to the localization algorithm.

Simulation results of FLC_i coefficients for 1% resistor tolerance and for capacitors tolerances 1%, 2.5%, 5% and 10% were drawn in Fig. 8. It is seen that the diagnosis method works correctly for 1% tolerances of all elements (for all elements $FLC_i > 98\%$) as also shown in Table 3.

Table 3. FLC_i for 1% tolerances of R and 1% tolerances of C

	R1 and C1	**R2**	**R3**	**C2**	**C3**
***FLC*$_i$ [%]**	R1 - 99.998 C1 - 99.992	99.511	98.515	98.013	98.745

Obviously, when element tolerances grow, the FLC_i diminishes under an acceptable level (particularly for 5% and 10%: $FLC_i > 92\%$, $FLC_i > 64\%$ adequately). Basing on presented simulation results and the fact that at present resistors even with tolerances 0.01% and capacitors with tolerances less that 1% are available on the market, we can say that the presented fault diagnosis algorithm is robust against element tolerances.

6. CONCLUSIONS

The author's significant contribution is the proposal of a new fault diagnosis approach of single faults of passive elements of analog electronic circuits based on the node-voltage relation method. This approach was adapted and implemented in ANSI C code for self-testing of the analog part of an electronic embedded system controlled by a control unit represented by the microcontroller. The approach of self-testing consists of three parts:

- Measurements of the amplitude and phase of voltages in accessible nodes of the tested analog circuit using internal resources of the microcontroller.
- A new way of creation of the fault dictionary describing the nominal state of the tested circuit and containing indirect parameters representing respective circuit faults.
- A new fault detection and localization algorithm implemented with success in the 8-bit microcontroller.

Thus, the main advantages of this approach are: a very small size of the fault dictionary (occupying only $12 + I \cdot 4 = 36$ bytes for $I = 6$ elements) and the simple and fast fault detection and localization algorithm, where only integer operations on integer numbers are executed.

REFERENCES

[1] Yang Z. R., Zwolinski M., Chalk C. D.,Williams A. C.: "Applying A Robust Heteroscedastic Probabilistic Neural Network to Analog Fault Detection and Classification", IEEE Transactions On Computer-Aided Design of Integrated Circuits And Systems, Vol. 19, No. 1, pp. 142-151, January 2000.

[2] Alippi C., Catelani M., Mugnaini M.: "SBT Soft Fault Diagnosis in Analog Electronic Circuits: A Sensitivity-Based Approach by Randomized Algorithms", IEEE Transactions On Instrumentation And Measurement, Vol. 51, No. 5, pp. 1116-1125, October 2002.

[3] Wang P., Yang S.: "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, Vol. 52, No. 10, pp. 2118-2127, October 2005.

[4] Catelani M., Fort A., Alippi C.: "A fuzzy approach for soft fault detection in analog circuits", Measurement, Vol. 32, Issue 1, pp. 73-83, July 2002.

[5] Toczek W., Zielonko R., Adamczyk A.: "A method for fault diagnosis of nonlinear electronic circuits", Measurement, Vol. 24, Issue 2, pp. 79-86, September 1998.

[6] Robotycki A., Zielonko R.: "Fault Diagnosis of Analog Piecewise Linear Circuits Based on Homotopy", IEEE Trans. on Instrument And Measurement, Vol. 51, No. 4, August 2002.

[7] Liu R.: "Testing and diagnosis of analog circuits and systems", Van Nostrand Reinhold, New York 1991.

[8] Czaja Z.: "A diagnosis method of analog parts of mixed-signal systems controlled by microcontrollers", Measurement., Vol. 40, Issue 2, pp. 158-170, February 2007.

[9] Savir J., Guo Z.: "Test Limitations of Parametric Faults in Analog Circuits", IEEE Trans. on Instrument And Measurement, Vol. 52, No. 5, pp. 1444-1454, October 2003.

[10] Catelani M., Fort A.: "Fault diagnosis of electronic analog circuits using a radial basis function network classifier", Measurement, Vol. 28, Issue 3, pp. 147-158, October 2000.

[11] Czaja, Z.; Zielonko, R.; **"**Fault diagnosis in electronic circuits based on bilinear transformation in 3-D and 4-D spaces", IEEE Trans. on Instrument And Measurement, Volume 52, Issue 1, pp. 97-102, February 2003.

[12] Lo C. H., Wong Y. K., Rad A. B., Chow K. M.: "Fusion of qualitative bond graph and genetic algorithms: A fault diagnosis application", ISA Transactions, Vol. 41, Issue 4, pp. 445-456, October 2002.

[13] Czaja Z., Załęski D.: "Employing a fuzzy logic based method to the fault diagnosis of analog parts of electronic embedded systems". Proceedings of the IEEE Instrumentation and Measurement Conference, IMTC'2007, Warsaw, Poland, May 2007.

[14] Wang P., Yang S.: "Soft fault test and diagnosis for analog circuits", Proceedings of the ISCAS 2005, Vol. 3, pp. 2188 – 2191, Kobe, Japan, 23-26 May, 2005.

[15] Martens G., Dyck J.: "Fault identification in electronic circuit with the aid of bilinear transformation", IEEE Trans. on Reliability, No 2, May 1972, pp.99-104.

[16] Czaja Z., Zielonko R.: "On fault diagnosis of analogue electronic circuits based on transformations in multi-dimensional spaces", Measurement, Vol. 35, Issue 3, pp. 293-301, April 2004.

[17] Atmel Corporation: "8-bit AVR microcontroller with 16k Bytes In-System Programmable Flash, ATmega16, ATmega16L", PDF file, available at: *http://www.atmel.com*, 2003.

[18] Swerlein R. L.: "A 10 ppm accurate digital ac measurement algorithm", Proceedings of the NCSL Workshop and Symposium, pp. 17-36, Albuquerque, USA, August 1991.

[19] Czaja Z., Załęski D.: „Implementation of an input-output method of diagnosis of analog electronic circuits in embedded systems", Proceedings of the 10th IMEKO TC10 International Conference on Technical Diagnostics, Hungary, Budapest, pp. 145-150, 9-10 June, 2005.