Neural Networks for Multiple Fault Diagnosis in Analog Circuits

Alessandra Fanni, Alessandro Giua, Enrico Sandoli
Istituto di Elettrotecnica, Università di Cagliari, P. zza d’Armi, 09123 Cagliari, Italy
email: fanni@elettrol.unica.it, giua@ece.rpi.edu

Abstract

Fault diagnosis of analog circuits is a complex problem. The paper discusses how the features of neural networks of learning from examples and of generalizing may be used to solve this problem. In a detailed applicative example, we show how, given the voltages values in a set of test points, a network may be trained to recognize catastrophic single faults on a circuit part of a direct current motor drive. The network is then used to diagnose multiple faults on two and three components. In this case the network is generally able to detect at least one of the malfunctioning components, although less sharply than in the case of single faults.

1 Introduction

Fault diagnosis of analog circuits is a complex problem. Classical solutions require either a huge amount of calculation if parameter identification methods are used, or a great number of simulations of faulty conditions if fault dictionary methods are used [6,7].

Novel approaches based on Artificial Intelligence techniques have been developed in recent years to overcome the limitations of these classical methods. Among them, we recall qualitative reasoning and rule-based expert systems.

Systems based on qualitative reasoning [2,3,10] use a qualitative causal model of the device to diagnose. The model behavior is compared to the actual device behavior and the discrepancies trigger a diagnostic procedure. Although these methods suffer from lack of information that often leads to non unique solutions, they are able to handle incomplete knowledge of linear and non linear circuits. However, they need a model, albeit qualitative, of the system at hand, with consequent great computational cost if the system is very complex.

Rule based systems explicitly associate a symptom to a cause by means of heuristic rules. They are simple to use, but very difficult to create and maintain: knowledge formalization has proved to be the bottleneck of this technology. The feature of the neural networks of learning from examples and of generalizing [12] has suggested their use in the fault diagnosis of electrical circuits. In this way it is possible to avoid many of the problems that affect other methods.

- The symptom-cause correspondence is derived automatically during the training of the net, without requiring an explicit formalization. Note that there is a small price to pay for this: while an expert system is generally able to justify its deductions, a neural network is not [13,14].

- Since the network is capable of generalizing, we may limit the number of fault simulations to a small set of accurately chosen faults. The network should be able to recognize fault configurations not explicitly comprised in the training set [5]. This is a clear advantage over fault dictionary methods.

- It is possible to avoid the problems connected with the calculation of circuit parameters and in general to the modelization, because the neural network does not need any circuit model or schema.
This paper reports on the results of our group at the University of Cagliari (see also [11]). Other approaches to the use of neural networks for circuit diagnosis have also been published recently. Keagle et al. [5] discuss how networks trained to recognize single faults may be used to detect multiple faults. Tests are performed on a digital circuit consisting of nine logical gates affected by stuck-at-1 or stuck-at-0. The paper also presents results on the performance of the diagnostic system as a function of the network architecture.

Meador et al. [8] compare feedforward neural network performance with other classifiers: gaussian maximum likelihood and k-nearest neighbors. In each experiment a single parameter deviation fault on an operational amplifier circuit is considered. The classifiers must separate the input patterns corresponding to the correct behavior and to the faulty one. Results show that the neural network gives higher accuracy than the other classifiers.

Potier et al. [9] propose to use neural networks as part of a model-based expert system for diagnosing lumped parameter devices. The purpose of the net would be that of solving the equations ruling the behavior of the diagnosed device, modeled as a set of interconnected components.

Thompson et al. [13] consider the problem of diagnosing an IC board with approximately 60 components, both analog and digital. They use a backpropagation neural network with a modular structure, i.e., each part of the net recognizes a particular fault.

Totton and Limeb [14] use neural networks to diagnose a high volume circuit board, part of a digital telephone exchange. They observed from historical data that failures on four types of components account for more than 85% of all faults. This led them to construct a network whose four outputs signal the presence of a faulty component of a given type, i.e., the network does not pinpoint the faulty component but simply detects what type of component is faulty.

In this paper we consider backpropagation neural networks. During a training process, based on single fault simulations, the network is able to associate the corresponding fault configuration to a generic configuration of test point voltage values, pointing out the malfunctioning component. Once the network has been trained, we have estimated the ability of the network to recognize multiple faults due to two and three simultaneously malfunctioning components. The voltage configurations corresponding to these multiple faults need not be presented to the network during the training.

We have restricted in the present investigation the class of faults considered. The faults in analog circuits may be catastrophic faults, that cause a large and sudden variation of the circuit parameter values, or degradational, and permanent faults, associated to slight variations of the circuit parameter values from their nominal values [1]. Since statistics have shown that in the analog circuits 80-90% of faults are catastrophic [4], we chose to simulate faults of this kind, such as short circuits and open circuits between two terminals of a component.

2 Neural model

Consider a circuit with $n$ components and a given set of $m$ test points. We will use a three layer backpropagation neural net with $m$ inputs and $n$ outputs to diagnose the circuit.

The voltage of all test points is measured by an acquisition board during several simulations in the absence of faults, in the presence of an open circuit on a single bipolar component, and in the presence of a short circuit on a single bipolar component. Thus, for a circuit with $n$ bipolar components, it is necessary to run $2n + 1$ simulations. We also simulated faults on components with more than two terminals. As an example, in the circuit shown in Section 4, there are trimmers and operational amplifiers. We considered two possible faults on a trimmer (cursor stuck up and cursor stuck down) and just one single fault on an operational amplifier (it was made inoperative by feeding with exceedingly high voltage). In general, let $s$ be the number of the single faults taken into account; thus one needs to perform $s + 1$ simulations.

In order to obtain a single meaningful value to represent it, we take its mean value. We have also considered the use of different indices, such as the root-mean-squared value. However, in the tests we performed the use of the mean value gave the best results. Thus, during the $i$-th simulation we obtain the vector $x_i = (x_i(1) \cdot x_i(m))^T$, where $x_i(j)$ is the mean value of the voltage of test point $j$.

We are now ready to construct the $s + 1$ input-output patterns that will be used to train the backpropagation neural network for the diagnosis of the circuit. Each pattern is given by a pair $(\vec{z}, \vec{y})$. The vector $\vec{y}$ associated to each $\vec{z}$ is defined as follows:

$$y_k(k) = \begin{cases} 0 & \text{if component } k \text{ is short circuited during the } i-th \text{ simulation}, \\ 0.5 & \text{if component } k \text{ is not faulty during the } i-th \text{ simulation}, \\ 1 & \text{if component } k \text{ is an open circuit during the } i-th \text{ simulation}, \end{cases}$$

We modelled the faults on the trimmer as follows: a value 1 of the output node corresponds to a
cursor stuck up fault, and a value 0 to a cursor stuck down fault. We modelled the fault on the
operational amplifier assigning a value 0 to the corresponding output node.

Once the net has been trained, it may be used to perform the diagnosis of the circuit. One needs
to give to the net as input vector \( \mathbf{z} \) the mean values of the measured test point voltages. The net
will produce an output vector \( \mathbf{y} \), a value of \( y(k) \) close to 0 (resp., 1) will pinpoint a short circuit
(resp., open circuit) fault of component \( k \); a value close to 0.5 will denote that the component
is correctly functioning. Note that the value of \( y(k) \) may be any number in the range \([0,1]\) thus
expressing a fuzzy membership of the component to one of the three states. This suggests that
the network may be used to diagnose even parameter deviation faults [8].

It is also clear that although the net has been trained with the results of single fault simulations,
it is potentially able to diagnose multiple faults. In this case, two or more elements of \( \mathbf{y} \) will be
close to 0 or 1.

This is the basic outline of our system. We now add a few remarks concerning the structure of
the neural network and the training algorithm.

- We tested both linear and sigmoid functions as node transfer functions. The sigmoid gave
  the best results.

- The updating of the connection weights on backpropagation neural networks [12] depends
  on two coefficients: the learning coefficient \( \eta \) and the momentum term \( \beta \). We obtained
  the best results using a value of \( \beta \) which decreased during the training from 0.6 to 0.05 and a
  value of \( \eta \) which decreased from 0.8 to 0.1.

- Networks with 2 layers are not capable of generalization and thus cannot be used to diagnose
  multiple faults.

- Networks with 3 layers and 4 layers are capable of generalization and their performance is
  improved if direct connections between input and output nodes are allowed. For the circuit
  discussed in Section 4, the best results were obtained using a 3-layer network with 48 nodes
  in the hidden layer. By increasing the number of hidden nodes, one slightly increases the
  capacity of the network but this considerably increases its complexity as well.

3 Data pre-processing

The data acquired from the circuit during the simulations are pre-processed before being used to
construct the training patterns. This is a fundamental step that may dramatically increase the
performance of the diagnostic system.

3.1 Filtering

Although the voltage measured at a test point is a function of time during each simulation, we used
a single value (the mean value) to represent it. In a first approach we tried to keep the information
on the shape of the voltage signal: we sampled it a few times during a period (say 7 times) and
we gave the net 7 values per test point. Thus, the number of input nodes of the network was
7m. Unfortunately, we realized that in these conditions the network behaves as a filter and its
performance decreases.

The same problem of filtering was observed in the case where the distance between the input
vectors of different patterns is small. To improve separability between training patterns we scaled the
inputs in the interval \([-1,1]\). For the test point \( j \) let \( x_{\text{max}}(j) = \max(x_0(j), \ldots, x_l(j)) \) and
\( x_{\text{min}}(j) = \min(x_0(j), \ldots, x_l(j)) \). We associate a scaling function to test point \( j \) as follows:

\[
f_j(x) = \frac{2x - x_{\text{max}}(j) - x_{\text{min}}(j)}{x_{\text{max}}(j) - x_{\text{min}}(j)}.
\]

Finally, given a vector \( \mathbf{x} \), we will transform it into a vector \( \mathbf{z} \) to be applied effectively to the net,
where:

\[
\mathbf{z} = (f_1(x(1)), \ldots, f_m(x(m)))^T.
\]

A similar approach is also used in [14].

Special care is required in all cases where \( x_{\text{max}}(j) - x_{\text{min}}(j) < \varepsilon \) (here \( \varepsilon \) is a fixed tolerance),
i.e., in all cases where \( x_i(j) \) is constant for all \( i \). This means that the reading of the test point \( j \)
voltage is meaningless as far as regards the diagnosis. We may simply avoid considering it, thus
decreasing the number of input nodes and simplifying the structure of the network.
3.2 Undistinguishable and undetectable faults

Another problem encountered was the following. Consider two (or more) components, say \(k\) and \(k'\) in parallel in the circuit. Let \((\vec{x}_k, \vec{y}_k)\) and \((\vec{x}_{k'}, \vec{y}_{k'})\) be the training patterns corresponding to the simulation of a short circuit on component \(k\) and \(k'\) respectively. Clearly \(\vec{x}_k = \vec{x}_{k'}\). However, \(\vec{y}_k \neq \vec{y}_{k'}\). In fact \(y_k(k) = 0\) and all other elements of \(\vec{y}_k\) are 0.5, while \(y_{k'}(k') = 0\) and all other elements of \(\vec{y}_{k'}\) are 0.5. Thus we train the net on a one-to-many correspondence, that is clearly not a suitable classification function [13]. The same problem appears when we consider the open circuit faults of series components, or more generally, when we have two or more undistinguishable single faults, i.e., faults that produce the same voltage configuration at the available test points. A similar problem may arise when a fault is undetectable. In this case, the training patterns corresponding to the fault-free simulation and to the simulation of the undetectable fault have the same input vector but different output vectors. The presence of undetectable faults may have different causes.

- A component whose behavior is the same when faulty or functioning correctly. As an example, for all practical purposes the behavior of a reverse biased diode is the same when the diode is functioning well or when it is affected by an open circuit fault.
- The limited number of test points may not allow detection of an abnormal behavior of the circuit.

We point out that the ambiguity due to undistinguishable or undetectable faults does not depend on our diagnostic system but derives either from the behavior and topology of the circuit or from the choice of test points.

We chose to fix this problem in two steps.

A first coarse solution takes care of topologically undistinguishable faults. From an inspection of the circuit one makes a list of all sets of parallel components. Then, one needs to consider a single short circuit fault simulation for each set \(C_i\) of parallel components. There will be a single training pattern \((\vec{x}_i, \vec{y}_i)\) for such a fault. The vector \(\vec{y}_i\) is such that \(y_i(k) = 0\) for all \(k \in C_i\), while all other components have a 0.5 value. A dual procedure takes care of sets of series components.

There is a more general approach for removing from the training set conflicting patterns due to behaviorally undetectable and undistinguishable faults. The approach entails the pre-processing of all data acquired during fault simulation. We noticed that, due to measurement noise, different acquisitions for the same fault simulation produce different values of \(\vec{x}\), all contained in a neighborhood of radius \(d\). Thus, one may not distinguish between two different faults whose corresponding vectors \(\vec{z}\) are closer than \(d\). In this case one may proceed as follows.

1. Acquire \(q\) measurements during each of the \(s + 1\) simulations. Let \(\vec{x}_i^1, \ldots, \vec{x}_i^q\) be the \(q\) acquisitions for the simulation \(i\) (\(i = 0\) in the absence of fault; \(i > 0\) for all other faults).
2. Compute the averages: \(\vec{z}_i = (\vec{x}_i^1 + \ldots + \vec{x}_i^q)/q\) for each simulation. Let \(d = \max_{i,j}(||\vec{x}_i - \vec{x}_j||)\).
3. A fault \(i\) such that \(||\vec{z}_0 - \vec{z}_i|| \leq d\) will be considered undetectable. The corresponding patterns will be removed from the training set.
4. Any two (or more) faults \(i\) and \(i'\) such that \(||\vec{z}_i - \vec{z}_{i'}|| \leq d\) will be considered undistinguishable and will belong to the same fault class. A single pattern \((\vec{x}, \vec{y})\) will represent the class, where \(\vec{x} = (\vec{x}_i + \vec{x}_{i'})/2\) and \(\vec{y'}\) will account for all faults in the class, as already shown for topologically undistinguishable faults.

4 Experimental results

We present the results obtained with our system on the circuit in Figure 1, part of a DC motor drive. The same circuit has also been diagnosed in [3]. In the figure, the \(n = 12\) test points are marked by numbers within circles, while the \(n = 36\) components are labeled by numbers in square brackets. Input nodes \(I_2\) and \(I_3\) are connected to DC voltage generators; input node \(I_1\) is connected to a voltage generator which has a direct component and a sinusoidal component with a frequency of 10 Hz.

The measurements were collected through an "Analog Device" RTI-815-A acquisition board. The board has 32 channels in single-ended mode and was controlled by a PC-486.

As explained in Section 2, there are 70 single faults to consider on this circuit. In fact, the circuit is composed of 36 components but only one fault is simulated for each of the two operational amplifiers. Thus, the overall training set should consist of 71 training patterns — the first being
related to the circuit behavior in absence of fault. However, due to the presence of conflicting patterns, the network performs poorly when all 71 patterns are presented for learning.

Pre-processing of the data gave the following results:

- The following sets contain topologically indistinguishable faults: \{ 10s, 11s \}, \{ 14s, 15s \}, 23s, 24s, \{ 27s, 28s, 29s \}, \{ 34s, 35s \}, \{ 16s, 17s \}, \{ 30s, 31s \}. Here 15s represents a short circuit fault on component 10, 16s represents an open circuit fault on component 16, etc.

- The set of behaviorally undetectable faults is: \{ 3s, 6s, 10s, 11s, 12s, 14s, 17s, 18s, 19s, 21s, 22s, 23s, 24s, 25s, 27s, 28s, 29s, 30s, 31s, 32s, 33s, 35s \}.

- The following sets contain behaviorally indistinguishable faults: \{ 4s, 6s \}, \{ 10s, 11s \}, \{ 14s, 15s, 16s, 17s, 18s, 19s \}, \{ 23s, 24s \}, \{ 34s, 35s, 36s \}, \{ 34s, 36s \}.

The procedure shown in Section 3 gives a reduced training set consisting of 33 different patterns. We will present the results obtained with four different nets. Net \( N_1 \) was trained on the 33 patterns for 40,000 iterations. Net \( N_2 \) was trained on the 33 patterns for 120,000 iterations. Net \( N_3 \) was trained on 99 patterns (we have acquired three measurements for each simulation, thus increasing the number of training patterns) for 120,000 iterations. Net \( N_4 \) was trained on 99 patterns for 360,000 iterations.

We fixed two threshold values for the output nodes of the network. Outputs greater than 0.8 (less than 0.2) will certainly pinpoint an open (short) circuit on the corresponding component. We consider two uncertainty bands; we say that an open (short) circuit on a component is likely if the corresponding output node has a value in the interval \([0.7, 0.8]\) \([0.2, 0.3]\).

As an example of diagnosis, Figure 2 and Figure 3 show the output obtained giving as input vector to the network the voltage configuration corresponding to an open circuit fault on component 2 and that corresponding to a short circuit fault on component 26, respectively. Finally, Figure 4 shows how the network behaves when its input vector is the voltage configuration corresponding to the presence of both faults previously considered. This shows that the network is not only capable of recalling a learned association but also of generalizing.

Table 1 shows the performances of the four networks when diagnosing single faults. Note that we have acquired new measurements for the same 33 simulations used to construct the training set. As can be seen, almost all single faults are correctly recognized. In a few cases the network shows a fault only as likely; this happens for faults whose input patterns are relatively close to the input pattern of the fault-free simulation. No false alarms were ever detected in these cases.

Table 2 shows the performances of the four networks when diagnosing two simultaneous faults. We have considered 60 different pairs of faults (undetectable faults are not considered, of course).

Table 3 shows the performances of the four networks when diagnosing three simultaneous faults. We have considered 28 different sets of faults.
Figure 2: Network output when diagnosing an open circuit on component 2.

Figure 3: Network output when diagnosing a short circuit on component 26.

Figure 4: Network output when diagnosing a multiple fault: open circuit on component 2 and short circuit on component 26.
### Table 1: Diagnosis of single faults (33 simulations).

<table>
<thead>
<tr>
<th>Faults correctly detected</th>
<th>N1</th>
<th>N2</th>
<th>N3</th>
<th>N4</th>
</tr>
</thead>
<tbody>
<tr>
<td>without false alarms</td>
<td>26</td>
<td>31</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>with false alarms</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Faults detected as likely</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>without false alarms</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>with false alarms</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Faults not detected</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>without false alarms</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>with false alarms</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

### Table 2: Diagnosis of two simultaneous faults (60 simulations).

<table>
<thead>
<tr>
<th>Two faults correctly detected</th>
<th>N1</th>
<th>N2</th>
<th>N3</th>
<th>N4</th>
</tr>
</thead>
<tbody>
<tr>
<td>without false alarms</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>with false alarms</td>
<td>9</td>
<td>16</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>One fault correctly detected</td>
<td>24</td>
<td>19</td>
<td>23</td>
<td>17</td>
</tr>
<tr>
<td>without false alarms</td>
<td>12</td>
<td>17</td>
<td>17</td>
<td>24</td>
</tr>
<tr>
<td>with false alarms</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>No fault correctly detected</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

### Table 3: Diagnosis of three simultaneous faults (28 simulations).

<table>
<thead>
<tr>
<th>Three faults correctly detected</th>
<th>N1</th>
<th>N2</th>
<th>N3</th>
<th>N4</th>
</tr>
</thead>
<tbody>
<tr>
<td>without false alarms</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>with false alarms</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>One or two faults correctly</td>
<td>8</td>
<td>2</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>detected</td>
<td>16</td>
<td>18</td>
<td>18</td>
<td>21</td>
</tr>
<tr>
<td>without false alarms</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>with false alarms</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
5 Conclusions

We have shown how a neural network, trained to recognize catastrophic single faults, may be used to diagnose multiple faults on analog circuits. In general we observed that the network is always able to learn and recall the single fault patterns presented during the training. Multiple faults on two and three components may also be diagnosed, although less sharply than in the single fault case, due to the presence of false alarms. In this case, however, since the network is generally able to detect at least one of the malfunctioning components, we may use an incremental repair procedure, substituting the faulty components one by one.

We plan to increase the performance of the diagnostic system by appropriately choosing test points and voltage supplies in order to reduce the number of undetectable and undistinguishable faults.

The research will then continue in its attempt to verify the performance of the adopted technique in the case in which the training set is based on the simulation of deviation faults, i.e., percentile variation of the circuit parameters with respect to the nominal value.

References


