Fault Diagnosis of Analog Circuits with Tolerances By Using RBF and BP Neural Networks

K. Mohammadi, A. R. Mohseni Monfared and A. Molaei Nejad

College of Electrical Engineering
Iran University of Science and Technology
Narmak, Tehran, Iran
mohammadi@iust.ac.ir

Abstract - This paper presents a method for analog circuit fault diagnosis by using neural networks. This method exploits DC approach for constructing dictionary in fault diagnosis by neural networks classification capability. In addition, Radial basis function (RBF) and backward error propagation (BEP) networks are considered and compared for analog fault diagnosis. The primary focus of the paper is to provide robust diagnosis using a mechanism to deal with the problem of component tolerance and reduce testing time. Simulation results show that the radial basis function network with reasonable dimension has double precision in fault classification but its classification is local, and backward error propagation network with reasonable dimension has single precision in fault classification but its classification is global.

1. Introduction

For more than three decades, the analog fault diagnosis has been of interest to researchers in circuits and systems. The research areas include computational complexity, automatic test pattern generation (ATPG), and design for test [3]. The analog circuit fault location can be an extremely difficult problem. This is because of the difficulty of measuring current without breaking connections, the lack of good fault models for analog components similar to the stuck-at-one and stuck-at-zero fault models, which are widely accepted by the digital test community, component tolerances and nonlinearities [1],[3],[4]. Generally, component tolerances make the parameters of circuit elements uncertain and the computational equations of traditional methods complex. The non-linear characteristic of the relation between the circuit and its constituent elements makes it even more difficult to diagnose faults on-line and may lead to false diagnosis. To overcome these problems, a robust and fast fault diagnosis method taking tolerances into account is needed.

Artificial neural networks (ANNs) have been applied in many areas such as pattern recognition, signal and image processing, etc. ANNs have the advantages of large-scale parallel processing, parallel storing, robust adaptive learning, and on-line computation. They are ideal for fault diagnosis of analog circuits with tolerances [2].

The research presented here exploits the robust classification capabilities of ANNs with fault dictionary approach to provide fault diagnosis of analog circuits with tolerances while minimizing computation costs. This is an extension of the results presented in [1],[2]. In addition to, we compare two neural network architectures, RBF and BPNN, for analog fault diagnosis. Section 2 introduces the ANNs. Section 3 discusses the analog circuit diagnosis method and outlines the steps involved in the development of the diagnosis system and examples are presented in section 4. Finally, conclusions are given in section 5.

2. Artificial Neural Networks

In recent years, ANNs have received great attention in many aspects of scientific research and have been applied successfully in various fields such as chemical processes, digital circuitry, control systems, etc. ANNs provide a mechanism for adaptive pattern classification. Even in unfavorable environments, they can still have robust classification. It should be stressed that choosing a suitable ANN architecture is vital for the successful application of ANNs [2]. Ever architecture of ANNs is suitable for a special application and has different precision compare to other architectures.

Based on learning strategies, ANNs fall into two categories: supervised and unsupervised. The BPNN is a supervised network but RBF is a network that can has supervised and unsupervised learning simultaneously. BPNNs can have different layers but typical BPNNs have two or three layers and RBFs have two layers of interconnecting weights. Fig.1 shows a two-layer network.

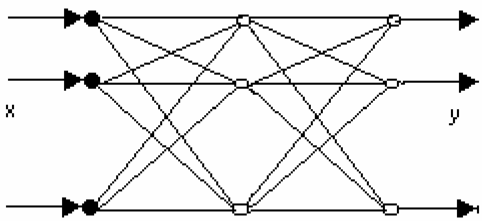

Fig. 1 A two-layer network

Each input node is connected to a hidden layer node and each hidden node is connected to an output node in similar way.

2.1 BPNN Algorithm

In this algorithm, learning takes place during the propagation of input patterns from the input nodes to the output nodes. The outputs are compared with the desired target values and an error is produced. Then the weights are adapted to minimize the error. The relation of output $o_i^{(l)}$ and input $o_j^{(l-1)}$ of layer j is defined as:

$$o_i^{(l)} = fs[I_i^{(l)}] \quad (1)$$

$$I_i^{(l)} = \sum w_{ij}^{(l)} o_j^{(l-1)} \quad (2)$$

Equation (1) can be transformed into:

$$F_s(I) = 1/(1+\exp(-I)) \quad (3)$$

The initial values of weights are assumed to be zero; and the weight between the j^{th} neuron of the $(k-1)^{th}$ layer and the i^{th} neuron of the k^{th} layer is defined as $w_{ij,k}$. The weight adaptation equation is given by

$$w_{ij,k}(t_n) = w_{ij,k}(t_{n-1}) - \alpha E(t_n)/w_{ij,k}(t_{n-1}) \bullet \Delta w_{ij,k}(t_{n-1})$$

Where $0<\alpha<1$, $0<\eta<1$ and $E = 1/2\sum(y_i - b_i)^2$, i=1…n, y_i is i^{th} output (4).

2.2 RBF Algorithm

The transformation from the input space to the hidden-unit space is nonlinear, whereas the transformation from the hidden-unit space to the output space is linear. Two examples of hidden layer function are follow:

1. Inverse multiquadrics

$$\varphi(r) = \frac{1}{(r^2+c^2)^{1/2}} \text{ For some } c>0, \text{ and } r\geq 0 \quad (5)$$

2. Gaussian functions

$$\varphi(r) = \exp(-\frac{r^2}{2\sigma^2}) \quad \text{For } \sigma>0, \text{ and } r\geq 0 \quad (6)$$

There are two important parameters in RBF neural network as follow:

1. Selection the center of hidden layer function.
2. Adjusting the weights of output layer.

There are different learning strategies that can follow in the design of an RBF network, as follow:

2.2.1 Fixed centers selected at random

The simplest approach is to assume fixed radial basis functions defining the activation functions of the hidden units. Specially, the locations of the centers may be chosen randomly from the training data set. A (normalized) radial-basis function centered at t_i is defined as:

$$G(\|x-t_i\|^2) = \exp(-\frac{M}{d^2}\|x-t_i\|^2) \quad , \quad i=1,\ 2,\ldots m \quad (7)$$

Where M is the number of centers and d is the maximum distance between the chosen centers. In effect, the standard deviation of all the Gaussian radial basis functions is fixed at

$$\sigma = \frac{d}{\sqrt{2M}} \quad (8)$$

Such a choice for the standard deviation σ merely ensures that the Gaussian functions are not too peaked or too flat; both of these extremes are avoided.

The only parameters that would need to be learned in this approach are the linear weights in the output layer of the network. A straightforward procedure for doing this is to use the pseudo inverse method (Broomhead and Lowe, 1998). Specifically, we have

$$W = G^{+} \mathbf{d} \quad (9)$$

Where **d** is the desired response vector in the training set.

2.2.2 Self-Organized selection of centers

In the second approach, the radial-basis functions are permitted to move the locations of their centers in a self-organized fashion, whereas the linear weights of the output layer are computed using a supervised learning rule .

2.2.3 Supervised selection of centers

In the third approach, the centers of the radial-basis functions and all other free parameters of the network undergo a supervised learning process; in other words, the RBF network takes on its most generalized form. For more information, refer to [6].

3. Fault Diagnosis of Analog Circuits

The usual method of automatically testing digital networks compares failed-board output levels with a set of prestored outputs on the Automatic Test Equipment (ATE). Similar techniques are developed for fault location of analog networks. We apply DC approach for dictionary construction. This method uses the dc voltages at the nodes of the circuit under arbitrary dc stimulus to constructing dictionary. The approach is summarized in the following steps:

Step 0
The test engineer provides the network description, fault definition and the input stimuli. Comment
The input stimuli are selected to exercise the "on", "off", and "linear" states of the semiconductor devices (e.g., diodes, transistors…).

Step 1
Different fault situations (single, hard, or soft faults) are inserted one at a time into the circuit simulator. The simulator computes dc nodal voltages and component overstresses resulting from the faults.

Step 2
Form the ambiguity sets for every measurement node using the different input stimuli.

Step 3
Manipulate the ambiguity sets to find out the level (degree) of isolation and the unnecessary measurement nodes.

Step 4
Construct the fault dictionary using the reduced set of measurement nodes. Indicate the ambiguity groups and the secondary overstresses caused by faults.

After constructing fault dictionary, we use it for training the neural network.

4. Examples

In this section, we consider two examples for checking the proposed method and comparing two described neural network architectures.

4.1 Resistive Circuit

First circuit is shown in Fig.2 [2]. There are 8 resistors. The nominal value of each resistor

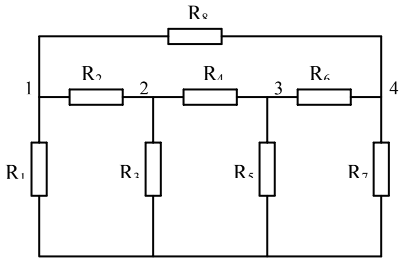

Fig.2 A resistive circuit

is 1Ω, and each element has a tolerance of ±5%. According to the topology of the circuit, three testing nodes are selected, which are numbered nodes 1,3, and 4. Thus, the neural networks should have 3 input nodes in the input layer and 8 output nodes in the output layer. The used BPNN have three layers which the first hidden layer has 16 neurons and the second hidden layer has 12 neurons and output layer has 8 neurons, therefore we have 36 neurons totally. Programmed in Neural Network Toolbox of MATLAB software, the BPNN algorithm and nodal voltage equations have been simulated by computer. After over 20,000 times of training and when the error performance is less than 0.007, the training of the BPNN is completed and the knowledge of the sample features is stored in it. The neural network is ready for checking now. Table 1 shows the results of checking it.

From Table 1, it can be found that diagnosis result is correct. However, it has a few error in the results (R_7 =1.06, output value is 0.75) and

Table 1 Results of BPNN in fault diagnosis of resistive circuit

Faulty element	Output node value relation to faulty element	Max. Value in the other output nodes
R_1 =5 Ω	1.0022	0.0810
R_7 =1.06 Ω	0.7553	0.1520
R_3 =0.8 Ω	0.8012	0.0980
R_5 =0.1 Ω	0.9056	0.0837
R_4 =50 Ω	1.0001	0.0051

the number of training is very large (20,000 times).

The used RBF network has 401 neurons in hidden layer and 8 neurons in output layer. It was formed one time after running the program. It must be noted that the training set has 401 vectors, which any resistor has a value inside rang [0-5 Ω]. Therefore, the number of training vectors equivalent to the number of input layer neurons. On this basis for covering the global classification [e.g. 0-1000 Ω] we need very large neurons in input layer. Table 2 shows results of checking the RBF network.

Table 2 Results of RBF in fault diagnosis of resistive circuit

Faulty element	Output node value relation to faulty element	Max. value in the other output nodes
R_1 =5 Ω	1.0000	0.0000
R_7 =1.06 Ω	0.9500	0.0500
R_3 =0.8 Ω	1.0021	0.0001
R_5 =0.1 Ω	1.0000	0.0000

From Table 2, it can be found that the diagnosis result is very accurate but it covers the rang [0-5 Ω] only and needs more neurons for global classification.

4.2 Active Circuit

Second circuit is shown in Fig.3 [5]. It is a simple JFET amplifier. There are 5 elements in this circuit, which have the nominal values as R_{G1}=1.4MΩ,R_{G2}=0.6MΩ,R_d=2.7kΩ,R_s=2.7kΩ, $V_{p(JFET)}$ =-4volts,$I_{DSS(JFET)}$=12mA, and each element has a tolerance of ±10%. We apply four testing nodes, which are numbered nodes 1,2,3,4. Thus, the neural network should have 4 input nodes in the input layer and 6 output nodes in the output layer.

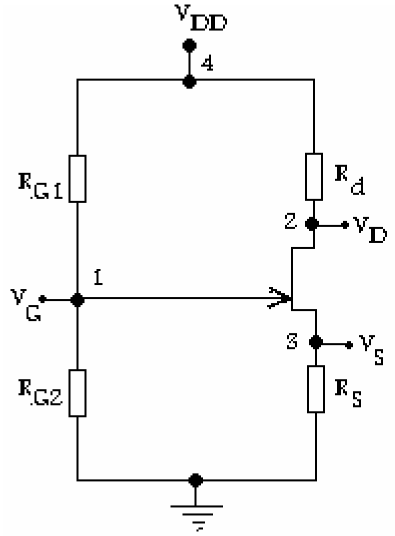

Fig. 3 Active circuit

According to step 0 of dictionary construction, we must select the input stimulus to exercise the “on”, “off”, and “linear” states of the JFET. The regions of operation of the transistor are defined as:

$V_{GS} < V_p \rightarrow$ JFET is OFF

$V_{GS} > V_p \rightarrow$ JFET is ON $\rightarrow \begin{cases} V_{GD} > V_p & \text{Triode} \\ V_{GD} < V_p & \text{Pinch off} \end{cases}$

With solving the operation equation of JFET and the circuit, under nominal condition, we get operation regions of the transistor as:

$V_{DD} < -13.3$ volts $\longrightarrow$ OFF

$-13.3 < V_{DD} < 15.4$ volts $\longrightarrow$ Triode

$15.4 < V_{DD} \longrightarrow$ Pinch off

We selected some stimulus of any JFET operation regions to constructing the fault dictionary. Also for any elements all, its possible faulty values are selected in every JFET operation regions.

With this basis, we constructed a fault dictionary with 1200 vectors. We are ready for training the neural networks now.

The used BPNN have three layers which the first hidden layer has 18 neurons, the second hidden layer has 12 neurons and output layer has 6 neurons, therefore we have 36 neurons totally. Programmed in Neural Network Toolbox of MATLAB software, the BPNN and nodal voltage equations have been simulated by computer. After over 30000 times of training and when the error performance is less than 0.01, the training of the BPNN is completed and the knowledge of the sample features is stored in it. The BPNN is ready for checking now. Table 3 shows the results f checking it.

Table 3 Results of BP in fault diagnosis of Active circuit

Faulty element	Output node value relation to faulty element	Max. Value in the other output node
$R_{G1} = 1$ MΩ	0.922	0.100
$R_s = 4$ kΩ	0.901	0.131
$I_{DSS} = 15$ mA	0.801	0.231
$V_p = -5$ v	0.770	0.257
$R_d = 2$ kΩ	0.815	0.235

From Table 3, it can be found that diagnosis results is correct, but it has a few error in the results (V_p= -5 v, output value is 0.770) and the number of training is very large (30,000 times).

The used RBF network has 1200 (is equivalent to the number of training vectors) neurons in hidden layer and 6 neurons in output layer. It was formed one time after running the program. Table 4 shows results of checking the RBF network

Table 4 Results of RBF in fault diagnosis of Active circuit

Faulty element	Output node value relation to faulty element	Max. Value in the other output nodes
$R_{G1} = 1$ MΩ	1.001	0.001
$R_s = 4$ kΩ	1.010	0.001
$I_{DSS} = 15$ mA	0.950	0.005
$V_p = -5$ v	0.931	0.009
$R_d = 2$ kΩ	0.950	0.004

From Table4, it can be found that the diagnosis results are very accurate but although RBF network has 1200 neurons in hidden layer its classification is local.

It must be noted that in this example we have two ambiguity groups (first group for R_{G1} and R_{G2} and second group for V_p and I_{DSS}) but we don't considered them.

5. Conclusions

A method that exploits DC approach for constructing dictionary in fault diagnosis by neural networks classification capability is proposed. Simulation results for two examples show that this method is robust to component tolerances and requires small after-test computation time.

Two neural network architectures, BPNN and RBF networks, for fault diagnosis are described and applied. Simulation results show that the radial basis function network with reasonable dimension has double precision in fault classification but its classification is local, and backward error propagation network with reasonable dimension has single precision in fault classification but its classification is global. In addition, time interval the training of BPNN is very larger than the RBF network. Therefore, RBF network is better in fault classification if special faults are considered and BPNN network is better in fault classification if all faults are considered.

REFERENCES

[1] J. W. Bandler and A. E. Salama, "Fault diagnosis of analog circuits," *Proc. IEEE*, vol. 73, pp. 1279-1325, Aug. 1985.

[2] Y. Deng, Y. He and Y. Sun, "Fault Diagnosis of Analog Circuits with Tolerances Using Artificial Neural Networks," *Circuits and Systems, 2000. IEEE APCCAS 2000. The 2000 IEEE Asia-Pacific Conference*, pp. 292-295.

[3] R. Spina and S. Upadhyaya, "Linear Circuit Fault Diagnosis Using Neuromorphic Analyzers," *IEEE Transactions on Circuits and Systems-II*: *Analog and Digital Signal Processing*, Vol. 44, No. 3, March 1997, pp. 188-196.

[4] T. Sorsa and H. N. Koivo, "Application of artificial neural networks in process fault diagnosis," *Automatica*, vol. 29, no. 4, pp. 843-849, 1993.

[5] A. D. Sedra and K. C. Smith, *MICROELECTRONIC CIRCUITS*, CBS College Publishing,1982.

[6] S. Haykin, *Neural Networks*, Macmillan College Publishing Company, Inc. 1994.