Fault Diagnosis of Analog Circuit Using DC Approach and Neural Networks

Ajeet Kumar Singh¹, Ajay Kumar Yadav², Mayank Kumar³
¹ M.Tech, EC Department, Mewar University Chittorgarh, Rajasthan, INDIA
² Asst. Professor, College of Engineering and Rural Technology Meerut, INDIA
³ M.Tech, EC Department, Mewar University Chittorgarh, Rajasthan, INDIA

Abstract
Several approaches for the diagnosis of faults in analog circuits have reasonably accuracy which comes at the cost of heavier processing requirements and lowered efficiencies. In this paper, we propose a time domain based technique for fault diagnosis using specifications extracted from the step response of a circuit. With the help of neural network we have to go for back propagation learning phase, and all the simulated results from multisim simulation i.e. node voltages of the circuit. We have implemented this on back propagation algorithm and after training network must have proper performance target. In neural-network based fault diagnosis, network works as a fault dictionary.

Keywords: Fault Diagnosis, DC Approach, Neural Networks.

2. INTRODUCTION

2.1 Fault location techniques
There are two main categories, approximation techniques that obtain a fault probability index related to each component and techniques that estimate system parameter including the faulty ones, from a reduced number of measurements. Both approaches have limited accuracy depending on the estimation that is used. [1]
Most of the traditional testing techniques for analog circuits have relied on specification testing, in which some or all response parameters are checked for conformity to the design specifications,[2] i.e. the performance functionalities of an analog module are measured and tested against set limits. Specification testing is costly in terms of test hardware and test time, and does not provide any information about the cause of failure. Therefore, there has been considerable research into devising alternate test methods for analog circuits. Fault-based test strategies are being increasingly used,[3] techniques test for the presence (or absence) of physical manufacturing-related defects (faults), thereby providing a quantitative estimate of the effectiveness and completeness of the testing process.[4] They also allow the test procedure to test only for the most likely group of faults induced by a manufacturing process.

Figure 2.1 Fault location techniques

When the number of measures is sufficient the parameters of the system values can be calculated using parameter identification techniques,[5] these values are obtained from the circuit measurements after simulation. The number of parameters that can be identified in a particular circuit is called the testability.

2.2 Fault verification techniques
This assumes that there are a limited number of measures in the sense that there are not enough to determine all the network parameters. They use network theory and mathematical theory to check the consistency of the equation set after simulation. The inconsistency implies a fault in one or more of the components related to the equation. Logical relations between inconsistency conclusions give the faulty components. The decomposition for large circuits can be classified in this group. [6]
2.3 Artificial intelligence techniques
Due to increase in circuit complexity system malfunction, detection and isolation are becoming more difficult. The following diagram represents ai techniques.
1- Traditional approach: these are the most common techniques used in the industry and are based on heuristics. The expert experience is collected in a rule base or as a decision tree. These approaches have been well tested and they are very simple and understandable. On the other hand, faults that are not predicted in advance will not be detected.[7]
2- Model based approaches make use of the model to predict fault in the real circuit. Its main disadvantage is its inability to deal with unsimulated faults and expert knowledge acquisition when causal models are needed.

3. FAULT DIAGNOSIS APPROACHES
In recent years, many methods of fault diagnosis in analog circuits have been proposed. A popular categorization of these methods is in the simulation before test (SBT) and simulation after test approach (SAT). Both the techniques are represented separately as following.[8]

3.1 Simulation before Test Approach (SBT)
The SBT approach is based on the comparison of the circuit responses associated with predefined test stimuli with those induced by different fault conditions. A subsequent classification must be considered to solve the fault detection and isolation problems. Simulation before test is also called fault dictionary approach or fault simulation.[9]

3.2 Simulation after Test Approach (SAT)
This approach is named simulation after test (SAT) fault isolation is obtained by estimating the circuit parameters from the measured circuit outputs. The identification of the circuit parameters is based on the assumption that enough information is available in the measurements and that measurements are mutually independent. The method suffers from several drawbacks associated with the identification procedure if the system is nonlinear or local minima issues arise. Both the techniques are used by different methods and depending on the circuit configuration. In addition, simulation after test techniques are generally time consuming when applied to large circuits.[10]

4. DC APPROACH TECHNIQUE
4.1 Flow chart for construction of fault dictionary
The construction of the dictionary is initiated by choosing the input signals to the circuit, the domain of analysis, and the responses to be measured. General-purpose computer-aided design simulator, to compute the dc voltages at the nodes of the circuit under arbitrary dc stimulus. The following diagram represents the preparation of fault dictionary with the use of this flow chart. And this flow chart is used for preparation of fault dictionary representation. For preparation of fault dictionary we have to take the node voltages for different kind of characteristics of the circuit i.e. fault or non-faulty and after that we will prepare the fault dictionary.

5. BASICS OF NEURAL NETWORK
Artificial neural networks have emerged from the studies of how brain performs. The human brain is made up of many millions of individual processing elements, called neurons, that are highly interconnected. Information from outputs of the other neurons, in the form of electrical pulses, are received by the cell at interconnections called synapses.[8] The synapses connects to the cell inputs, dendrites, and the single output of the neurons appears at the axon. An electrical pulses is sent down the axon when the total input stimuli from all the dendrites exceeds a certain threshold.

Artificial neural network are made up of simplified individual models of the biological neurons that are connected together to form a network. Information is stored in the network often in the form of different connection strengths, or weights associated with the synapses in the artificial neuron model. Some of the various properties of neural networks are, Inherent parallelism in the network architecture due to the repeated use of capability of learning information by example.[10] The learning mechanism is often achieved by appropriate adjustment of the weights in the synapses of the artificial neuron models.
1-Ability to generalize to new inputs.
2-Robustness of noisy data that occurs in real world application.
3-Fault tolerance, in general network performance does not significantly degenerate if some of the network connections become faulty.

5.1 Neuron model
A neuron with a single scalar input and no bias and with bias appears in the following figure.

The scalar input \( p \) is transmitted through a connection that multiplies its Strength by the scalar weight \( w \) to form the product \( wp \), again a scalar. Here \( wp \) is the only argument of the transfer function \( f \), which produces the scalar output \( a \).

The transfer function net input \( n \), again a scalar, is the sum of the weighted input \( wp \) and the bias \( b \). This sum is the argument of the transfer function \( f \). Here \( f \) is a transfer function, typically a step function or a sigmoid function that takes the argument \( n \) and produces the output \( a \).

Examples of Various transfer functions are in “Transfer Functions” Note that \( w \) and \( b \) are both adjustable scalar parameters of the neuron. The central idea of neural networks is that such parameters can be adjusted so that the network exhibits some desired or interesting behavior. Thus, we can train the network to do a particular job by adjusting the weight or bias parameters, or perhaps the network itself will adjust these parameters to achieve some desired end.

The various transfer functions are given as,
1- Hardlim, the hard limit transfer function shown above limits the output of the neuron to either 0, if the net input argument \( n \) is less than 0, or 1, if \( n \) is greater than or equal to 0.

2- logsigmoid, the sigmoid transfer function shown below takes the input, which can have any value between plus and minus infinity, and squashes the output into the range 0 to 1. This transfer function is commonly used in back propagation networks, in part because it is differentiable.

The scalar input \( p \) is transmitted through a connection that multiplies its Strength by the scalar weight \( w \) to form the product \( wp \), again a scalar. Here \( wp \) is the only argument of the transfer function \( f \), which produces the scalar output \( a \).

The scalar input \( p \) is transmitted through a connection that multiplies its Strength by the scalar weight \( w \) to form the product \( wp \), again a scalar. Here \( wp \) is the only argument of the transfer function \( f \), which produces the scalar output \( a \).

3- The following figure illustrates the linear transfer function.
4- The following figure illustrates the tansig transfer function

![Tansig Transfer Function](image)

**Figure 5.5 Pure linear transfer function**

**Figure 5.6 Tansig transfer function**

### 5.2 Artificial neuron model

A neuron with a single R-element input vector is shown below. Here the individual element inputs $p_1, p_2, \ldots, p_R$ are multiplied by weights $w_1, 1, w_1, 2, \ldots, w_1, R$ and the weighted values are fed to the summing junction. Their sum is simply $Wp$, the dot product of the (single row) matrix $W$ and the vector $p$.

The neuron has a bias $b$, which is summed with the weighted inputs to form the net input $n$. This sum, $n$, is the argument of the transfer function $f$.

$N = w_1, 1p_1 + w_1, 2 + \ldots + w_1, R \cdot p_R + b$

The net input to the transfer function $f$ is $n$, the sum of the bias $b$ and the product $Wp$. This sum is passed to the transfer function $f$ to get the neuron's output $a$. The output $a$ is obtained by summing the weighted inputs to the neuron and passing the result through a non-linear activation function, various non-linear functions are possible and some of these e.g., hard limiter, threshold, tanh, sigmoidal functions.

### 5.3 Network Architectures

Two or more of the neurons shown earlier can be combined in a layer, and a particular network could contain one or more such layers. First consider a single layer of neurons. A network can have several layers. Each layer has a weight matrix $W$, a bias vector, and an output vector. To distinguish between the weight matrices, output vectors, etc. for each of these layers in our figures, we append the number of the layer as a superscript to the variable of interest.

### 5.4 A Layer of Neurons

A one-layer network with R input elements and S neurons follow

- **Input**: $p_1, p_2, \ldots, p_R$
- **Weights**: $w_{1,1}, w_{1,2}, \ldots, w_{1,R}$
- **Bias**: $b_1$
- **Net Input**: $N = \sum w_{1,i}p_i + b_1$
- **Output**: $a = f(N)$

**Figure 5.8 One-layer network**

### 6. MULTISIM SIMULATION ANALYSIS

We have taken up the Sallen-Key band pass filter circuit to illustrate this approach. The Sallen-Key band pass filter, shown in Fig. is a second order active filter, which is greatly appreciated for its simplicity of design. Its transfer function is obtained as:

![MULTISIM SIMULATION ANALYSIS](image)

**Figure 6.1 Experimental circuit for fault diagnosis**

Various simulated results are taken from multisim simulation DC analysis given in fig.

www.ijaert.org
7. CONCLUSIONS
The example presented in the circuit for the active circuit show the effectiveness of the proposed approach in locating various types of faults. The main features of this method are:

a) It is applicable to small & large circuits.

b) AC & DC extreme and drift faults may be located under the proper current excitation and corresponding voltage measurements.

c) The simulated result is trained with back propagation neural network.

d) The computation is minimal and the diagnosis is directly carried out from the neural network.

REFERENCES


