JOURNAL OF ELECTRONIC TESTING: Theory and Applications 17, 29–36, 2001

Fault Diagnosis of Analog Circuits Using Bayesian Neural Networks with Wavelet Transform as Preprocessor

FARZAN AMINIAN
Trinity University, 715 Stadium Dr., San Antonio, TX 78212, USA
farzan@engr.trinity.edu

MEHRAN AMINIAN
St. Mary's University, One Camino Santa Maria, San Antonio, TX 78228, USA
mehran@quantum.stmarytx.edu

Received April 19, 2000; Revised October 10, 2000

Editor: M. Soma

Abstract. We have developed an analog circuit fault diagnostic system based on Bayesian neural networks using wavelet transform, normalization and principal component analysis as preprocessors. Our proposed system uses these preprocessing techniques to extract optimal features from the output(s) of an analog circuit. These features are then used to train and test a neural network to identify faulty components using Bayesian learning of network weights. For sample circuits simulated using SPICE, our neural network can correctly classify faulty components with 96% accuracy.

Keywords: analog fault diagnosis, neural networks, analog circuits, Bayesiasn learning

1. Introduction

Fault diagnosis of digital circuits has reached the point of automation while analog circuits still rely heavily on engineers' intuition and experience to develop test strategies. Analog fault diagnosis is challenging because of component tolerances, nonlinear effects and poor fault models. Neural networks present a promising approach to perform analog fault diagnosis since they do not require an in-depth study or any model for these effects. Instead, faulty behavior is intentionally introduced in the circuit and training data is collected from the circuit output(s). Features (signatures) extracted from the circuit output(s) together with the fault classes are then presented to the neural network as input-output pairs during the training phase. The neural network then adjusts its weights such that its outputs are relatively close to the ideal outputs (desired fault classes) for all the training data. The adjustment of weights is usually governed by the minimization of a predefined error goal which is a measure of the difference between the neural network and ideal outputs. Once the error goal is reached, training stops and the ability of the neural network to generalize is evaluated by presenting previously unseen features. If the neural network correctly determines the fault classes associated with these features, training has been successful. Otherwise, training will resume.

Application of neural networks to analog fault diagnosis is studied in references [1, 3, 7]. The training process in these works is based on the maximum likelihood technique which assigns a single set of weight values to the trained network by minimizing the error goal. A review of these works shows the importance of preprocessing the output(s) of the circuit under test to generate optimal features for training the neural network. Preprocessing leads to a minimal set of features to distinguish among fault classes reducing neural network's size significantly while improving its performance. In this work, we apply Bayesian networks to perform fault diagnosis of analog circuits. The Bayesian approach assumes a probability distribution function for the network weights based on the training data observed and uses this distribution function to evaluate network output(s). The motivations for using Bayesian approach are active learning, model comparison and feature relevance [2]. Active learning refers to the ability of Bayesian networks to provide guidance on which area of the feature space data should be collected so that it is most informative. Model comparison and feature relevance allow for different neural network architectures to be compared based on training data only and determination of relative importance of different features used to train the neural network. Our fault diagnostic system first preprocesses the output(s) of an analog circuit by wavelet decomposition [8], normalization and principal component analysis (PCA) [2] to extract optimal features. These optimal features are then used to train and test the neural network. Bayesian neural networks are still in their infancy and, as a result, have had limited applications in different areas. The purpose of this work is to present an approach for training these networks to perform analog fault diagnosis.

The material in this work is arranged in the following order. In Section 2, we briefly discuss the proposed preprocessing techniques used for feature extraction. Section 3 covers Bayesian learning for classification problems. In Section 4, we present the sample circuits and their associated faults used in our study and Section 5 discusses the results of this work.

2. Preprocessing and Feature Extraction

To extract features for training the neural network, the output of the circuit under test first undergoes wavelet transform to generate its wavelet coefficients. These wavelet coefficients fall into two categories of approximations and details which measure the low and high frequency components of the output signal, respectively. Since the basic structure of a signal is captured in its approximation coefficients [1, 5], they represent appropriate features for analog fault diagnosis. Next, normalization is applied to rescale input features to avoid large dynamic ranges in one or more dimensions in the feature space. This ensures that large variations in feature sizes do not dominate more important but smaller trends in the data. In this work, all features are normalized to have a zero mean and unity standard deviation. Finally, PCA is employed to minimize the dimensionality of the feature space (inputs to the neural network) by selecting a minimal set of approximation coefficients as features that remain distinguishable across fault classes. These preprocessing techniques significantly reduce the neural network architecture and improve its performance. The optimal features selected by our preprocessing techniques to train the neural network are the first coefficients of approximation levels one through five generated by the Haar wavelet function [1]. The feature selection process is covered in detail in Ref. [1]. Section 5 briefly discusses the role of features in distinguishing among fault classes to provide insight into the performance of our proposed neural network.

3. Bayesian Network Architecture

The Bayesian network used in our study has as many modules as there are fault classes. Each module is assigned to one fault class and has an output that estimates the probability that input features belong to that class. Since the output of each module is to be interpreted as probability, it is appropriate for the single neuron in its output layer to have the logistic sigmoid activation function [2]. To train a module, features extracted from a circuit output are paired with a target value of one if they belong to the fault class associated with that module. Otherwise, the target value would be zero. Test features are then assigned to the fault class with the highest probability as estimated by the outputs of the modules. As shown in Fig. 1, this approach assigns the original problem involving *m* fault classes to Bayesian modules that diagnose individual fault classes. The Bayesian learning process for a typical module in our fault diagnostic system is discussed next.

In Bayesian learning [2], we define a prior probability distribution function for the weights, $P(\mathbf{w})$, in the absence of training data. This distribution function is usually taken to have the form

$$P(\mathbf{w}) = \frac{1}{(2\pi/\alpha)^{W/2}} \exp\left(-\frac{\alpha}{2}\sum_{i=1}^{W} w_i^2\right). \tag{1}$$

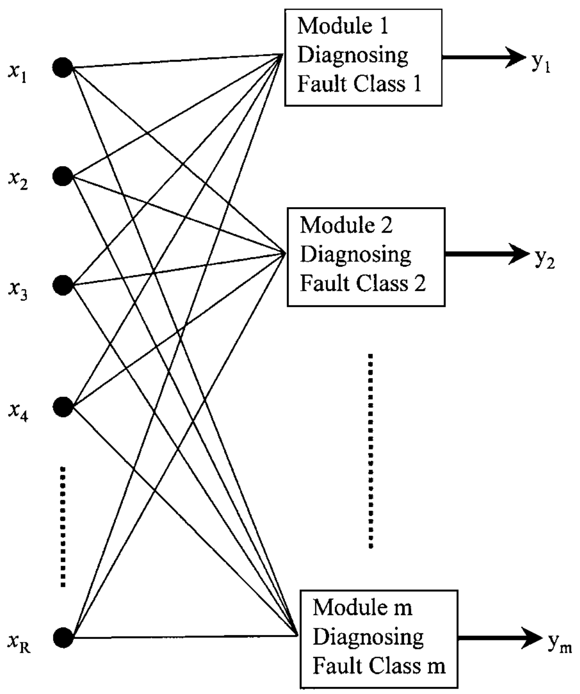

Fig. 1. The assignment of m fault classes to Baysian neural network modules that diagnose individual fault classes. Each module has an output estimating the probability that the input feature $\mathbf{x} = (x_1, x_2, \ldots, x_R)$ belongs to its associated fault class.

In this equation, $\mathbf{w} = (w_1, w_2, \ldots, w_W)$ represents the network weights and α is a parameter controlling the shape of the prior distribution function. This parameter is chosen such that $P(\mathbf{w})$ is broad to imply that in the absence of training data, many weight vectors are equally likely to be taken by the network. Once the training set $D = (\mathbf{x}_1, t_1), (\mathbf{x}_2, t_2), \ldots, (\mathbf{x}_N, t_N)$ is observed, the posterior weight distribution can be found from Bayes' theorem according to the equation

$$P(\mathbf{w} \mid D) = \frac{P(D \mid \mathbf{w}) P(\mathbf{w})}{P(D)}. \tag{2}$$

In this equation, the likelihood function $P(D \mid \mathbf{w})$ represents the probability of observing the training set given a network weight vector $\mathbf{w}$ and $P(D)$ is the normalization constant representing the probability of the training set D. In a typical module k $(1 \leq k \leq m)$, the probability of observing either target values ($t_i = 0$ or 1) for a training pair $(\mathbf{x}_i, t_i)$ is

$$p(t_i \mid \mathbf{w}) = y_k(\mathbf{x}_i; \mathbf{w})^{t_i(k)} [1 - y_k(\mathbf{x}_i; \mathbf{w})]^{1-t_i(k)}, \tag{3}$$

Where $y_k(\mathbf{x}_i; \mathbf{w})$ represents the module's output for feature vector $\mathbf{x}_i$ and weight vector $\mathbf{w}$ (see Fig. 1)[1]. Hence, the likelihood of observing the N pairs associated with the training set takes the form

$$P(D \mid \mathbf{w}) = \prod_{i=1}^{N} y_k(\mathbf{x}_i; \mathbf{w})^{t_i(k)} [1 - y_k(\mathbf{x}_i; \mathbf{w})]^{1-t_i(k)}. \tag{4}$$

Using Eqs. (1), (2) and (4), The posterior distribution can now be written as

$$\begin{aligned} P(\mathbf{w} \mid D) &= \frac{1}{Z} \exp\left(-G - \alpha \sum_{i=1}^{W} w_i^2\right) \\ &= \frac{1}{Z} \exp(-S(\mathbf{w})), \end{aligned} \tag{5}$$

where $G(D \mid \mathbf{w}) = -\ln(P(D \mid \mathbf{w}))$ and Z is the normalization constant. The posterior distribution given by Eq. (5) tends to be narrow in weight space to imply that some weight vectors are more compatible with the observed training set. We can now compute the probability that an input feature vector $\mathbf{x}$ belongs to the class k associated with a module using the posterior distribution of the network weights:

$$\begin{aligned} P(k \mid \mathbf{x}, D) &= \int P(k \mid \mathbf{x}, \mathbf{w}) P(\mathbf{w} \mid D)\, d\mathbf{w} \\ &= \int y_k(\mathbf{x}; \mathbf{w}) P(\mathbf{w} \mid D)\, d\mathbf{w}. \end{aligned} \tag{6}$$

In this equation, $P(k \mid \mathbf{x}, \mathbf{w})$, which is the probability that a feature vector $\mathbf{x}$ belongs to the class k for a given weight vector $\mathbf{w}$, is approximated by the module output $y_k(\mathbf{x}; \mathbf{w})$. Since the posterior weight distribution is narrow in weight space, it can be approximated by a Gaussian centered on the maximum posterior weight vector $\mathbf{w}_{MP}$ [6] to get

$$P(\mathbf{w} \mid D) = \frac{1}{Z^*} \exp\left[-S(\mathbf{w}_{MP}) - \frac{1}{2} \Delta\mathbf{w}^T \mathbf{A} \Delta\mathbf{w}\right], \tag{7}$$

where $\Delta\mathbf{w} = \mathbf{w} - \mathbf{w}_{MP}$, $\mathbf{A} = \nabla\nabla S|_{\mathbf{w}=\mathbf{w}_{MP}}$, and Z^* is the normalization factor appropriate for the Gaussian distribution. It is important to note that the most probable weight vector $\mathbf{w}_{MP}$ corresponds to the minimum of the error function $S(\mathbf{w})$ defined in Eq. (5) which can be obtained using any optimization technique. Using the Gaussian approximation for the posterior weight distribution, we now compute $P(k \mid \mathbf{x}, D)$ using Monte Carlo techniques to evaluate the integral in Eq. (6) [4]. If a set of M sample vectors $\mathbf{w}_i$ are generated from the Gaussian distribution for $P(\mathbf{w} \mid D)$, we can approximate Eq. (6) as

$$P(k \mid \mathbf{x}, D) = \frac{1}{M} \sum_{i=1}^{M} y_k(\mathbf{x}; \mathbf{w}_i). \tag{8}$$

Evaluation of $P(k \mid \mathbf{x}, D)$ from this equation for every module determines which fault class a test feature vector $\mathbf{x}$ belongs to. In summary, to train a module, we first need to find the maximum posterior weight vector $\mathbf{w}_{MP}$ from Eq. (5). This corresponds to the minimum of $S(\mathbf{w})$ which can be found using an optimization technique such as Quasi-Newton method [2] used in this work. Next, a Gaussian approximation to the posterior weight distribution $P(\mathbf{w} \mid D)$ is assumed and sample weights generated from this distribution are used to evaluate Eq. (8). A test feature vector $\mathbf{x}$ is then assigned to the class with the highest probability.

Our proposed neural network architecture has several important properties:

1. Since every module is assigned to one fault class only, it has a simple architecture (very few adjustable parameters) and a fast training time.
2. Since all the modules are completely independent of each other, they can be trained in parallel to reduce the training time by the number of modules.
3. If the modules are trained sequentially, the training time increases in proportion to the number of faults. However, the training time for every module, and therefore the overall neural network, are very short.
4. Our proposed neural network architecture is very reliable in detecting indistinguishable fault classes. It also performs very well in distinguishing among fault classes that have significant overlap in one or more feature values. The reliability of our neural network in dealing with such classes stems from the fact that the outputs of its modules represent probabilities of fault classes.

These properties are further discussed and demonstrated in Section 5 where we present our results.

4. Sample Circuits and Faults

In order to evaluate the effectiveness of our Bayesian neural network in analog fault diagnosis, we have applied it to two sample circuits shown in Figs. 2 and 3. Figure 2 shows a Sallen-Key band pass filter with the nominal values of all components leading to a central frequency of 25 kHz. We assume all resistors and

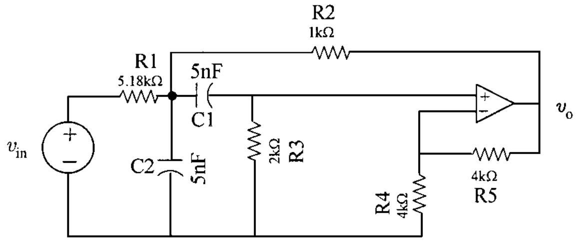

Fig. 2. A 25 kHz Sallen-Key band pass filter used in our study.

capacitors have standard tolerances of 5% and 10%, respectively. Faulty behaviors are assumed to result from R3, C2, R2 and C1 being 50% higher or lower than their nominal values. In order to generate training and testing data from the circuit output, we set faulty components in the SPICE simulations and vary other components within their standard tolerances. This results in nine fault classes for the filter including the no-fault (nft) class. Figure 3 shows a 10 kHz two-stage four-op-amp biquad low pass filter. The faulty components used in our study are $C1\Uparrow$, $C2\Uparrow$, $C3\Uparrow$, $C4\Uparrow$, $R16\Uparrow$, $R17\Downarrow$, $R19\Uparrow$, $R21\Downarrow$, $R3\Uparrow$, $R4\Downarrow$, $R6\Uparrow$, $R7\Downarrow$, $R8\Uparrow$ and $R9\Uparrow$ where $\Uparrow$ and $\Downarrow$ imply significantly higher and lower than nominal values. The values of the fourteen faulty components associated with this filter are shown in Table 1. In all our SPICE simulations, we have taken the filter input to be a single pulse of height 5 V and a duration of 10 μs. This input pulse is much

Table 1. Fault classes used for the two-stage four-op-amp Biquad circuit. The nominal and faulty values are also specified.

Fault class	Nominal	Faulty value
C1$\Uparrow$	0.01 μF	0.051 μF
C2$\Uparrow$	0.01 μF	0.02 μF
C3$\Uparrow$	0.01 μF	0.048 μF
C4$\Uparrow$	0.01 μF	0.031 μF
R16$\Uparrow$	1800 Ω	7800 Ω
R17$\Downarrow$	4840 Ω	1600 Ω
R19$\Uparrow$	10000 Ω	30000 Ω
R21$\Downarrow$	10000 Ω	3750 Ω
R3$\Uparrow$	2200 Ω	5000 Ω
R4$\Downarrow$	1570 Ω	600 Ω
R6$\Uparrow$	10000 Ω	16500 Ω
R7$\Downarrow$	10000 Ω	5500 Ω
R8$\Uparrow$	440 Ω	2200 Ω
R9$\Uparrow$	2640 Ω	4300 Ω

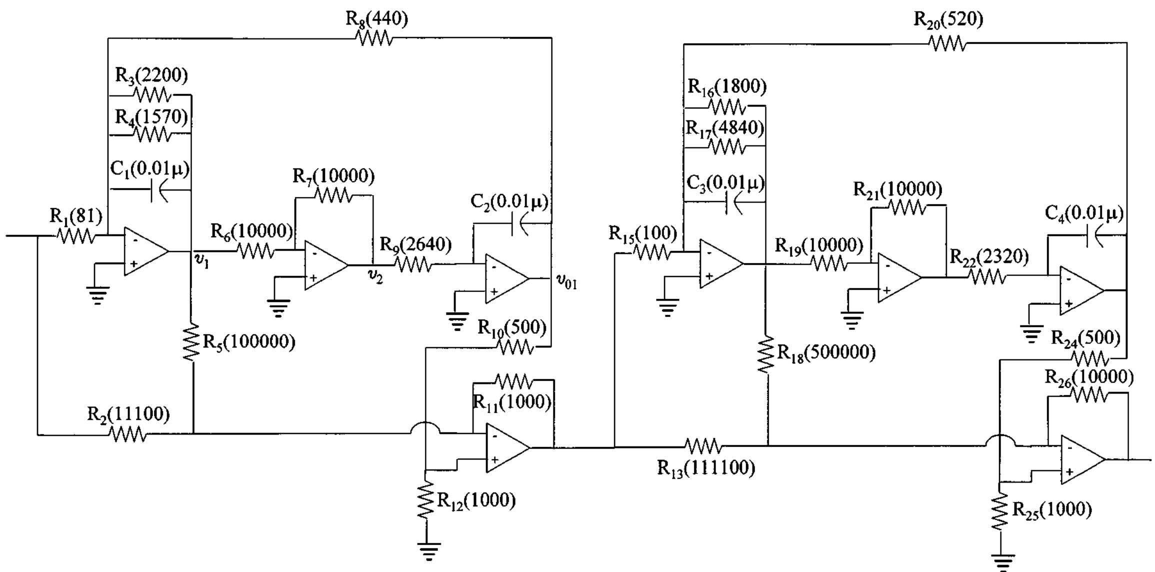

Fig. 3. The two-stage four-op-amp biquad low pass filter used in our study. All resistors are in ohms.

narrower than the inverse of the filter's bandwidth resulting in an output that well approximates its impulse response. As discussed in Section 2, the features extracted from these outputs are then used to train the neural network to classify faulty components. It is important to note that our fault diagnostic system can be applied to any fault of interest in a circuit. As long as features extracted from the circuit output(s) remain distinguishable across fault classes, a neural network can be trained to perform fault diagnosis. In the next section, we discuss the performance of the Bayesian neural network in classifying the faults associated with our sample circuits.

5. Results

To diagnose the faults associated with the band pass filter, our fault diagnostic system requires a neural network with nine modules. Each module consists of a single neuron only with a logistic sigmoid transfer function and five inputs corresponding to the features selected by the preprocessing techniques. Since every module has only six adjustable parameters (five weights and a bias), it has a very simple architecture and a short training time. To train the neural network for fault diagnosis of the Sallen-Key band pass filter, we have used a training set of size 54 for each fault. The trained network can correctly classify 97% of the test data with a size of 27 per fault class. The training time for each neural network module is a few seconds on a 333 MHz Dell workstation. The accuracy of the neural network in classifying the test data indicates that all nine fault classes are distinguishable by its input features. This can be illustrated using Fig. 4 which shows the range of values, measured by the mean and one standard deviation, for the five features. Each plot in this figure corresponds to one feature with its range obtained from a data set of size 20 per fault class. The x-axis in each plot represents the fault classes in the order $C1\Uparrow$, $C1\Downarrow$, $C2\Uparrow$, $C2\Downarrow$, nft, $R2\Uparrow$, $R2\Downarrow$, $R3\Uparrow$ and $R3\Downarrow$ and the y-axis corresponds to feature values. An examination of this figure shows that all fault classes are distinguishable by one or more features. For instance, features 1 and 4 (counted row-wise from left to right) can separate $R2\Uparrow$ and $R2\Downarrow$ fault classes for the band pass filter or $C1\Downarrow$ and $R3\Uparrow$ fault classes are distinguished by all features.

The two-stage four-op-amp biquad low pass filter is more complex and has fourteen fault classes shown in Table 1. The range of values for the five features using a sample of size 20 per fault class are shown in Fig. 5. In this figure, the x-axis shows the fault classes in the same order as Table 1 and the y-axis represents the feature values. This is a more challenging fault diagnosis problem for the neural network since there are indistinguishable fault classes. Moreover, some fault classes are not separated well in all feature values making their

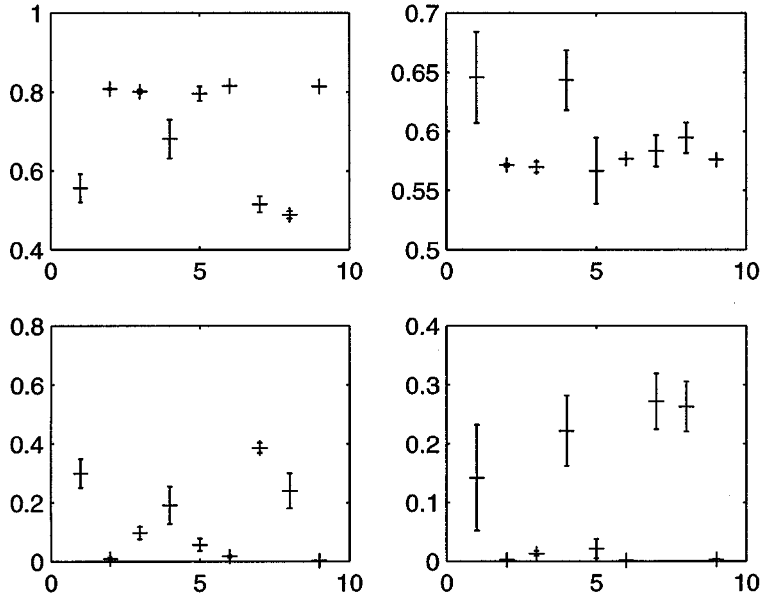

Fig. 4. Range of values (average plus one standard deviation) associated with the features representing faults of the Salley-Key band pass filter. Each plot corresponds to one feature. The fault classes are shown in the order $C1\Uparrow$, $C1\Downarrow$, $C2\Uparrow$, $C2\Downarrow$, nft, $R2\Uparrow$, $R2\Downarrow$, $R3\Uparrow$, and $R3\Downarrow$ on the x-axis and the y-axis shows the feature values.

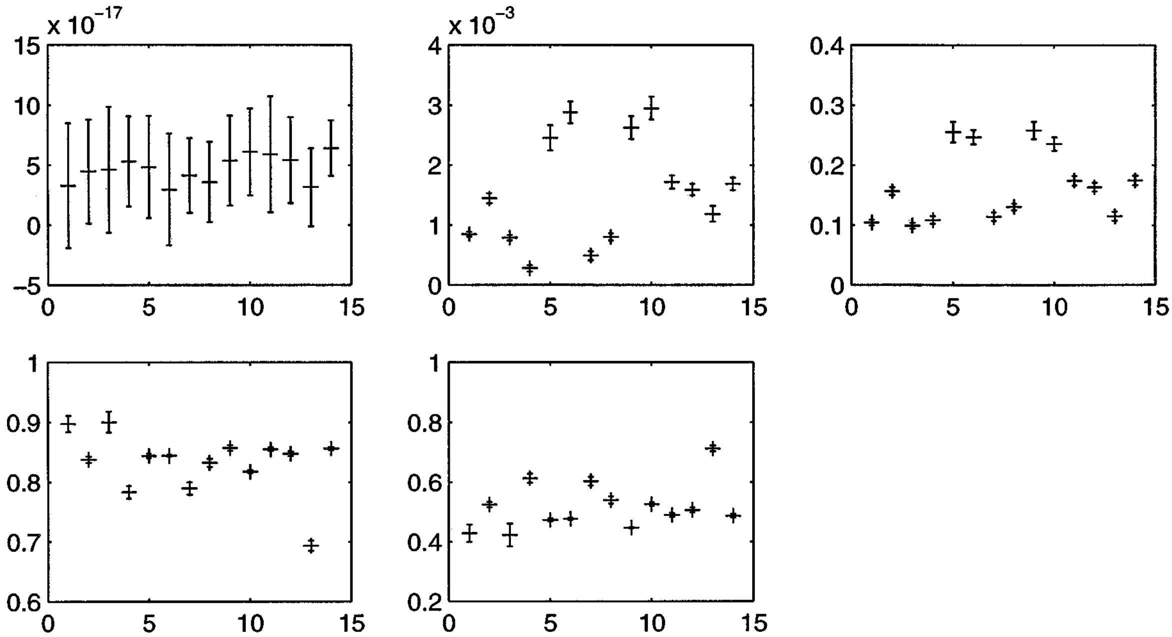

Fig. 5. Range of values (average plus one standard deviation) associated with the features representing faults of the two-stage four-op-amp biquad low pass filter. Each plot corresponds to one feature. The fault classes are shown in the order $C1\Uparrow$, $C2\Uparrow$, $C3\Uparrow$, $C4\Uparrow$, $R16\Uparrow$, $R17\Downarrow$, $R19\Uparrow$, $R21\Downarrow$, $R3\Uparrow$, $R4\Downarrow$, $R6\Uparrow$, $R7\Downarrow$, $R8\Uparrow$ and $R9\Uparrow$ on the x-axis and the y-axis shows the feature values.

fault diagnosis challenging. To train the neural network, we have used a training set of size 150 per fault class. The architecture for the fourteen modules associated with our neural network is identical to the one used for the band pass filter. Since each module is assigned to one fault class, there is no increase in its architecture complexity compared to the previous case. As a result, the training time for each module is still a few seconds. For complex circuits with a large number of faults, we expect the overall training time to increase in proportion with the number of modules. However, since the training time per module is of the order of a few seconds, the overall training time remains short. As indicated before, an alternative to the sequential training of the modules is to train them all in parallel which reduces the overall training time to that required by a single module. For both our sample circuits, the overall training time is under one minute when all modules are trained sequentially.

Our studies indicate that the Bayesian neural network can successfully diagnose all fault classes in Table 1 except $R6\Uparrow$, $R7\Downarrow$, and $R9\Uparrow$ which are found to be indistinguishable. When a test data belongs to any of these fault classes, the three modules associated with these faults take similar output values. This implies that it is equally likely for the test data to belong to any of these three fault classes making them indistinguishable. The reason our proposed neural network can reliably detect indistinguishable fault classes is that its outputs estimate the probabilities of fault classes. An inspection of Fig. 5 verifies that since the five features associated with these fault classes overlap completely, they can not be separated. This can be explained by an analysis of the low pass filter in Fig. 3 which shows

$$\frac{dv_{o1}}{dt} = \frac{R_7}{R_6 R_9 C_2} v_1. \tag{9}$$

The expression 9 remains relatively unchanged for the faulty component values shown in Table 1 making these fault classes indistinguishable. When these fault classes are placed in one ambiguity group, our neural network can correctly classify 96% of the test data with a size of 75 per fault class. The accuracy of the neural network in classifying the test data also indicates its ability to distinguish among fault classes that overlap in one or more feature values (e.g. $R16\Uparrow$ and $R17\Downarrow$). These results indicate that our neural network can reliably perform analog fault diagnosis using simple modules requiring a short training time.

6. Conclusion

We have applied Bayesian neural networks to analog fault diagnosis using wavelet transform, normalization and principal component analysis as preprocessors. Our approach breaks down the several fault classes present in a circuit into single-fault problems assigned to individual modules. Since each module needs to recognize one fault class only, it has a simple architecture, fast training time and superior performance. Application of our fault diagnostic system to two sample circuits result in 96% correct classification of test data. Our neural network has the important property that its outputs estimate the probabilities of fault classes. This enables the neural network to perform reliably in identifying indistinguishable fault classes or separating faults with overlapping feature values. Our results prove that Bayesian neural networks can accurately perform fault diagnosis of analog circuits.

Note

1. A training pair $(\mathbf{x}_i, t_i)$ consists of a feature vector extracted from the circuit output and its associated fault class. To train the module k with this pair, we use the notation $(\mathbf{x}_i, t_i(k))$ since the target output $t_i(k)$ changes from one module to another. For instance, if the feature vector $\mathbf{x}_i$ belongs to fault class 2, we have

$$t_i(k) = \begin{cases} 1 & k = 2 \\ 0 & k \neq 2 \end{cases}.$$

References

1. M. Aminian and F. Aminian, "Neural-Network Based Analog-Circuit Fault Diagnosis using Wavelet Transform as Preprocessor," *IEEE Transactions on Circuits and Systems II*, vol. 4, no. 2, February 2000.
2. C.M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, 1995.
3. M. Catelani and M. Gori, "On the Application of Neural Networks to Fault Diagnosis of Electronic Analog Circuits," *Measurement*, vol. 17, no. 2, pp. 73–80, 1996.
4. G.S. Fishman, *Monte Carlo Concepts, Algorithms, and Applications*, New York: Springer-Verlag, 1996.
5. B.B. Hubbard, *The World According to Wavelets*, A.K. Peters Ltd., 1996.
6. D.J.C. MacKay, "A Practical Bayesian Framework for Back-Propagation Networks," *Neural Computation*, vol. 4, no. 3, pp. 448–472, 1992d.
7. R. Spina and S. Upadhyaya, "Linear Circuit Fault Diagnosis using Neuromorphic Analyzer," *IEEE Transactions on Circuits and Systems II*, vol. 44, no. 3, 1997.
8. G. Strang and T. Nguyen, *Wavelet and Filter Banks*, Wellesley-Cambridge Press, 1996.

Farzan Aminian received the M.S. and Ph.D. degrees in electrical engineering from The Ohio State University in 1983 and 1989. He joined the Engineering Science Department at Trinity University in 1989 where he is currently serving as an associate professor. His areas of interest include applications of Monte Carlo technique, solid state devices and neural networks. Dr. Aminian is currently serving as the chairman of the Central Texas Section of IEEE.

Mehran Aminian received the M.S. and Ph.D. degrees in electrical engineering from the University of Oklahoma in 1982 and 1989. Currently, he is serving as an associate professor in the Engineering Department at St. Mary's University. Dr. Aminian's research interests are in quantum collision theory, quantum devices and neural networks.