A Neural Based Approach and Probability Density Approximation for Fault Detection and Isolation in Nonlinear Systems

P. Boi* and A. Montisci

Abstract. A locally recurrent neural network based fault detection and isolation approach is presented. A model of the system under test is created by means of a dynamic neural network. The fault detection is performed on the basis of the statistical analysis of the residual provided by the estimated density shaping of residuals in the case of nominal value of all the parameters, made of a simply neural network. The approach is illustrated by using the Rössler hyperchaotic system.

Keywords: Locally recurrent neural networks, nonlinear systems diagnosis, gradient-based training.

1 Introduction

In this paper we introduce a neural network based approach for fault detection and fault isolation of nonlinear circuits. Several methods have been proposed for fault diagnosis of analog circuits, board, and chip levels[1]-[18]

In [2] the authors propose an optimization based multifrequency test generation method for detecting parametric faults in linear analog circuits.

In [8] a decomposition approach for diagnosis and fault prediction in large analogue and switched capacitor circuits is proposed, reduced to an analysis of the corresponding nullator-norator models at a single test frequency. Algorithms for isolation of faulty nodes and subcircuits are presented. The topological conditions for diagnosing faulty nodes and elements are given, taking into account the element parameters tolerance and measurement errors.

In [10] a coefficient-based test (CBT) is introduced for detecting deviation faults in analog linear circuits. The method uses pseudo Monte-Carlo simulation to determine the transfer function coefficients for all the test points, and the circuit is declared faulty if any coefficients falls outside the fault free range.

In [5] a fault diagnosis system for analog circuit testing based on the Simulation Before Test (SBT) approach, is presented. Frequency response analysis is carried out of the circuit under test, then parameters are extracted such that they give unique values for every configuration of the circuit.

In[6] mathematical programming models are developed for the diagnosis of soft fault in analog circuit. At the same time, the tolerance question is handled, too.

The authors in [19] and [20] propose a Generalized Algorithm of Fault Tolerance (GAFT), using time, structural and information redundancy types. In [19] it is shown that

* Electrical and Electronic Eng. Dept - Univ. of Cagliari – Italy.

L. Iliadis and C. Jayne (Eds.): EANN/AIAI 2011, Part I, IFIP AICT 363, pp. 296–305, 2011.
© IFIP International Federation for Information Processing 2011

the design of fault tolerant systems is efficient if malfunctions are tolerated at the element level. The advantage of element malfunction tolerance is proven in reliability terms. Maximum reliability of fault tolerant system is achievable with less than a duplication system and depends on malfunction/permanent fault ratio and coverage of faults. System level of fault tolerance prerogative is the reconfiguration from permanent faults.

In [20] a scheme, is proposed to implement fault tolerant system software for routers using recovery points. It is shown that GAFT is capable of detecting, identifying, localizing faults, reconfiguring the hardware to achieve a repairable state and, if necessary, reconfiguring the software as well. Separation of malfunction and permanent fault tolerance is discussed in terms of their impact on system reliability.

In [22] is presented a fault diagnosis method for analog circuits that use a neural network classifier. The authors show that preprocessing based on the kurtosis and entropy of signals simplifies the network architecture, reduces the training time, and improves the performance of the network.

In the case of diagnosis of dynamic systems, we should to use dynamic neural networks to catch the behavior of the system. In particular, the Locally Recurrent Neural Network (LRNN) have the feed-forward multi-layer architecture and their dynamic properties are obtained using feedbacks inside the neurons, so that in spite of the neural network maintains the topological structure of a Multi Layer Perceptron, it can exhibit a dynamical behavior like recursive networks. A neural network can be trained to mimic the system behavior or to predict the next sample of the input. In the present work, a LRNN predictor has been trained, validated and tested using the input learning data, related to different operating conditions, that are collected from a simulation model of Rössler's system.

After finishing the training process, the neural network is ready for on-line residual generation. The residuals are generated by comparing the system output with the output of the neural model. A fault detection can be performed if a parameter deviation in the system under test determines a change in the dynamics of the diagnostic signal that appreciably affects the residual.

The decision on the occurrence of a fault (detection) is made when the modulus of all components of the output estimation error exceeds its corresponding threshold provided by the statistical distribution of the residual estimated in the case of nominal value of all the parameters.

We assume that a fault is said to be isolable if the fault isolation system is able to reach a correct decision in finite time. Faults are isolable if they are mutually different according to a certain measure quantifying the difference in the effects that different faults have on measurable outputs and on the estimated quantities in the isolation system.

The organization of the paper is as follows. In Section II locally recurrent neural networks are introduced. In Section III, the training algorithm is presented. Section IV, the diagnostic approach is presented. In Section V, density shaping using neural networks is explained. Section VI reports the experimental results referring to the benchmark of Rössler's system. Section VII reports conclusions.

2 The Neural Model

An artificial neural network used to model the system behavior belongs to the class of so-called locally recurrent globally feed-forward networks [23]. Its structure is similar to a multi-layer perceptron where neurons are organized in layers, but dynamic properties are achieved using neurons with internal feedbacks. The block structure of the *i-th* neuron considered is presented in Fig. 1. Thus, the i-*th* neuron in the dynamic network reproduces the past signal value with two signals: the input *u t* and its output y_k t. The weighted sum of inputs is calculated according to the formula:

$$s_i(k) = w_i u(k), \tag{1}$$

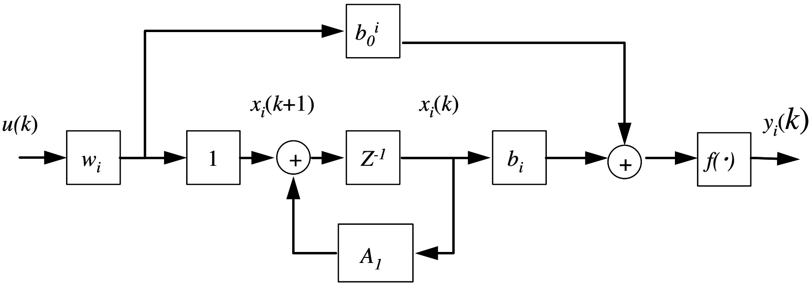

Fig. 1. ith locally recurrent neuron

$\boldsymbol{w}_i(k) = [w_1^i, w_2^i, ..., w_n^i]$ is the input weights vector, n is the number of inputs, $\boldsymbol{u}(k) = [u_1^i, u_2^i, ..., u_n^i]^T$ is the input vector (T-transposition operator). The weights $\boldsymbol{w}_i$ have the same role as the weights in the static feedforward networks. The states of the ith neuron in the network can be described by the following equation:

$$\boldsymbol{x}_i(k+1) = \boldsymbol{A}_i \boldsymbol{x}_i(k) + \boldsymbol{W}_i \boldsymbol{u}(k), \tag{2}$$

where $\mathbf{x}_i(k) = [x_1^i, x_2^i, ..., x_n^i]$ is the state vector. $\boldsymbol{W}_i = \mathbf{1}\boldsymbol{w}_i$ is the weight matrix ($\mathbf{1} \in \boldsymbol{R}^r$ and $\mathbf{1}$ is the vector of ones) and the matrix of state transition $\boldsymbol{A}_i$ has the form:

$$\boldsymbol{A}_i = \begin{bmatrix} -a_1^i & -a_2^i & \cdots & -a_{r-1}^i & -a_r^i \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \tag{3}$$

finally, the output of the neuron is described by

$$y_i(k) = f(\boldsymbol{b}_i \boldsymbol{x}_i(k) + \boldsymbol{d}_i \boldsymbol{u}(k), \boldsymbol{g}_i), \tag{4}$$

where $f(\cdot)$ may be a linear or a non linear activation function, $\boldsymbol{d}_i = \boldsymbol{b}_o^i \boldsymbol{w}_i$, $\boldsymbol{b}_i = [b_1^i, b_2^i, ..b_r^i]$ is the vector of feed-forward filter parameters. $\mathbf{g}_i = [g_1^i, g_2^i]$ is the vector of the activation function parameters consisting in bias and slope of the activation function.

3 Training Algorithm

All unknown network parameters can be represented by a vector $\boldsymbol{\theta}$ composed of elements $\boldsymbol{A}_i$, $\boldsymbol{W}_i$, $\boldsymbol{b}_i$, $\boldsymbol{d}_i$ and $\boldsymbol{g}_i$, where p is the number of network parameters. The main objective of training is to adjust the elements of the vector $\boldsymbol{\theta}$ in such a way as to minimize some loss function

$$\theta^* = \underset{\theta\in\Theta}{arg}\, min\mathbf{J}(\theta), \tag{5}$$

where $\boldsymbol{\theta}^*$ is the optimal network parameter vector Θ is the set of constraints defining the feasible values for the parameters. This network can be trained using a stochastic gradient-based algorithm [24] in the form:

$$\hat{\theta}_{k+1} = \hat{\theta}_k - lr_k \hat{\boldsymbol{g}}\left(\hat{\theta}_k\right) \tag{6}$$

where $\hat{\boldsymbol{g}}\left(\hat{\theta}_k\right)$ is the estimate of the gradient $\partial\boldsymbol{J}/\partial\hat{\theta}$ based on the measurements of $L(\cdot)$ the loss function $\boldsymbol{J}(\theta)$. The gradient estimate is obtained by

$$\hat{g}_{ki}(\hat{\theta}_k) = \frac{y_{1k} - y_{2k}}{2c_k\Delta_{ki}} \quad \forall i = \mathbf{1,\ldots,}\, p \tag{7}$$

where $y_{1k} = L(\hat{\theta}_k + C_k\boldsymbol{\Delta}_k) + \varepsilon_{1k}$, $y_{2k} = L(\hat{\theta}_k - C_k\boldsymbol{\Delta}_k) + \varepsilon_{2k}$, ($\varepsilon_{1k}$ and ε_{2k} represent measurement noise terms), $\mathit{\Delta}_k$ is a p-dimensional random perturbation vector that in our work is obtained by the symmetric Bernoulli distribution ± 1. lr_k and c_k are calculated as follows: $lr_k = a/(A+k)^{\alpha}$, $c_k = c/k^{\gamma}$.

4 Diagnostic Approach

In this paper a Fault Detection approach is proposed, which can be applied to both linear and nonlinear systems. First a neural network like that in Fig. 1 is trained off-line by means of the procedure described above in order to predict one step ahead the output signal of the system. The number of hidden neurons and the number of state variables are determined by a trial and error procedure. When the trained network is used to predict the signal, unavoidably the output will exhibit a certain prediction error or residual. On the basis of a validation signal, different from the training signal, the statistic distribution of the residual is estimated both in the case of nominal value of all the parameters and when one or more of them have a value within the tolerance range. This allows us to determine a threshold for the detection of the fault.

Finally, the residual ***r*** should be used by a classifier to determine both the type and occurrence time of possible faults. To evaluate residuals and to obtain information about faults, a simple threshold can be applied. If residuals are smaller than threshold value a the tested process is considered to be healthy, otherwise it is considered faulty. The Fault isolability analysis is made by introducing a fault mismatch function between any of two types of fault. Then two faults are reciprocally isolable if the fault mismatch function exceeds its corresponding threshold.

5 Density Estimation

A random vector ***x*** of an arbitrary distribution can be transformed into a new random vector ***y*** of a different distribution maximizing the mutual information between ***x*** and ***y*** [25]. This can be achieved by maximizing the entropy of the output of the neural network that has input ***x*** and output ***y*** .

$$h(y) = -E\{\mathbf{ln} f_y(y)\} = -E(\mathbf{ln} f_x(x)) + E(\mathbf{ln}|\partial y / \partial x|), \tag{8}$$

where the entropy of ***x***, $f_x(x)$, can be neglected because it may be considered unaffected by alterations in a parameter *w*. Let us consider the network of Fig. 2, it has only one input *x*, and the activation function *f(x)* is a sigmoidal function. The input probability density function is approximated by

$$q_x(x) = |\partial y / \partial x|, \tag{9}$$

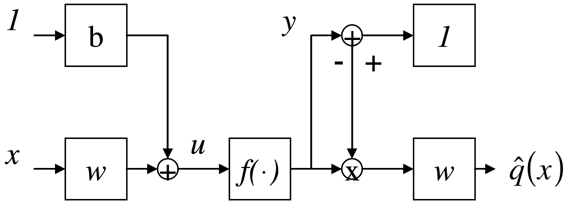

Fig. 2. Neural network model that estimates the density function

This network can be trained using the on-line version of stochastic gradient ascent rule [25] in the form:

$$\Delta v = \frac{\partial h(y)}{\partial v} = \frac{\partial}{\partial v}\left(\log\left|\frac{\partial y}{\partial x}\right|\right) = \left(\frac{\partial y}{\partial x}\right)^{-1} \frac{\partial}{\partial v}\left(\frac{\partial y}{\partial x}\right), \tag{10}$$

where *v* is the generalized network parameter. The equation (10) represents a self-organizing algorithm. The equation (10) contains the anti-Hebbian term, which keeps *y* away from saturation at 0 and 1. The learning rule for the weight parameter *w* is

$$\Delta w = \frac{1}{w} + x(1 + 2\mathrm{y}), \tag{11}$$

similarly the rule for the bias weight parameter is

$$\Delta b = \mathbf{1} - \mathbf{2y}, \tag{12}$$

After the training, the estimated probability density function can be calculated using the scheme shown in Fig. 2. The estimation of the input probability density function takes the form:

$$\hat{q}_x(x) = |w| y(\mathbf{1} - y) \tag{13}$$

Then we calculate the threshold corresponding to a given significance level α.

When a fault (a deviation of the parameter) occurs in the system, a residual should deviate from the level observed to the fault-free case. The quality of the fault detection system can be evaluated using a number of performance indexes. The sensitivity of the proposed fault diagnosis system in the fault-free case is checked using the *false detection rate* r_{fd}*=n°false detections/tot n° fault signals.* This index is used to check the system in the fault-free case. Its value shows a percentage of false alarms.

The *true detection rate* is: *rsd=n°successful detection/tot n° faults.* This index is used in the case of faults and describes the efficiency of fault detection.

The *time of fault detection* t_{dt} is a period of time needed for detection of a fault measured from the instant t_0 when the threshold is exceeded the first time. A fault is considered certain when the residues of all state variables used in the diagnostic system simultaneously exceed the threshold during the period $t_{dt.}$

Now, Let us assume that when a fault is detected at time $t=t_d$ the fault isolation system is activated.

5.1 Fault Isolability Analysis

We introduce the fault mismatch function between the *s-th* fault and the *r-th* fault:

$$h_{jsr}(t) = e_{js}(t) - e_{jr}(t). \qquad r, s = \mathbf{1} \ldots N, r \neq s \tag{14}$$

e_{js} and e_{jr} are the residuals of the *j-th* variable used in the isolation system if there are the faults *s* and *r* respectively, and *N* is the number of possible faults. Let us suppose that a fault *s* occurring at time $t=t_0$ is detected at time $t=t_d$. Then the fault *s* is isolable if, for each $r \in \{1,\ldots,N\} \setminus \{s\}$, there exist some time $t_i > t_d$ and at least one variable j used in the isolation system, such that

$$|h_{jsr}(t)| > h_o \tag{15}$$

where the threshold $h_0$is determined empirically, for each variable *j*. The sensitivity of the proposed fault isolation system is checked by calculating, for each $r \in \{1,\ldots,N\} \setminus \{s\}$, the *true isolation rate*:

r_{srj}=n°successful isolations/n°tot faults of type s. This index is used in the case of faults and describes the efficiency of fault isolation.

6 Experiments

The Rössler's hyperchaotic system [26] has been used to validate the diagnostic approach. The difficulty to interpret the residual signal is due to the fact that the trajectory in the state space critically depends on the initial state of the system, therefore one cannot define as a reference a fault-free behavior. The equations that describe the system are:

$$\begin{cases} \dot{x}_1 = -x_2 - x_3 \\ \dot{x}_2 = x_1 + ax_2 + x_4 \\ \dot{x}_3 = x_1x_3 + b \\ \dot{x}_4 = -cx_3 + dx_4 \end{cases} \tag{16}$$

Rössler's system has a hyperchaotic attractor when a=0.25, b=3, c=0.5, d=0.05. In order to show the suitability of the method, we tried to detect a parameter drift with respect to a point in the parameter space where the system exhibits a chaotic behavior, so that we haven't a reference behavior of the system without faults. In turn, residual evaluation is carried out by using statistical analysis discussed in Section 4. The fault diagnosis system is evaluated using several faulty conditions. An 1-5-2-1 network structure has been adopted, with 2 state variables in hidden neurons and a linear activation function. Such network is trained over 1000 samples in order to iteratively predict one step ahead three state variables (x_1, x_2, x_4) on the basis of the past samples, when all the system parameters are at their nominal value. The sum of squared errors calculated over 9000 testing samples is equal to 0.095. The reference value of the average and the variance of the residual is evaluated by performing the signal prediction being all the parameters at their nominal value but the initial state different from that one of the training signal.

Modelling results for 800 testing samples are shown in Fig. 3, where the model output is marked by the dotted line, and output of process by the solid line.

Using neural model of the process, for each system parameter, three residuals are calculated, corresponding to the prediction of three output variables. The same evaluation is performed when we decrease the parameter a of 5% and of 10% of its nominal value. The average and the variance of each residual are used as feature for the diagnosis. A greater number of experiments can aid to better describe the frontier decision for the fault detection, whereas in general further test points improve the detectability of faults. Residuals are used to train the neural network described in Section 4 to approximate the corresponding probability density function. The neural network in Fig. 2 is trained off-line for 90000 steps using an unsupervised learning (6). To perform a decision, a significance level $\alpha = 0.05$ is assumed. The test set consists of 100 signals each consisting of 1000 samples for each operating condition. In the fault-free case, the *false detection rate r_{fd}* is equal to *0.1.* In fault conditions, the results of fault detection are presented in Table 1. In each case, if the true detection rate (5) is close to 1 it means that detection of faults is performed surely. In order to perform decision about faults, and to determine detection time a time-window with the length $n = 5$ has been used. If during the following n time steps all residuals exceed the threshold then a fault is signalled. Application of time-window prevents the situation when a temporary true detection signals a fault. All faulty conditions can be classified. The results of fault detection are presented in Table 2.

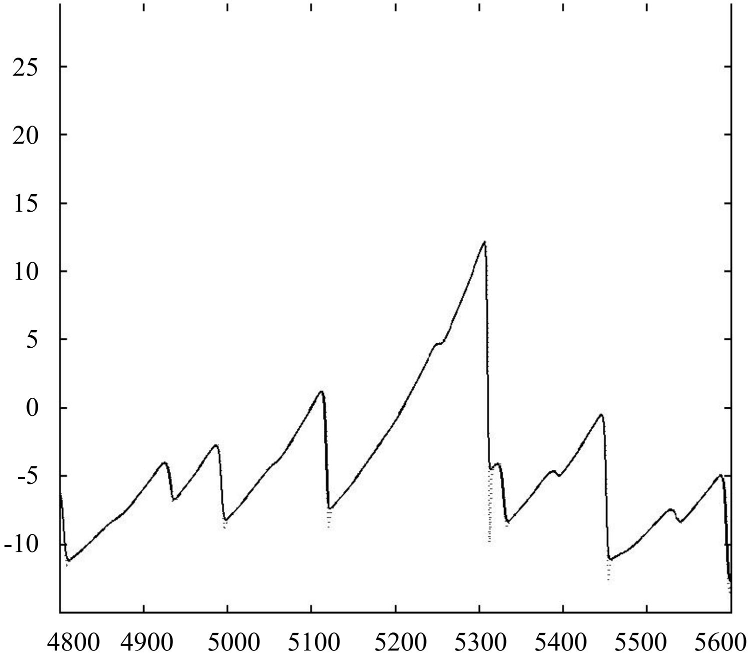

Fig. 3. Results of modelling

Table 1. Performance index for the fault conditions in to detection system when we decrease the parameter a of 5% (f1) and of 10%(f2)

	f1	f2
r_{sd}	0.7	0.78

Table 2. Performance indexes for the fault conditions in to isolation system

$r_{5\%,10\%}$	$r_{10\%,5\%}$
0.68	0.72

7 Conclusion

The purpose of this paper was to propose a model-based fault detection and fault isolation system for nonlinear circuits. The model of the system under test is realized by means of a locally recurrent neural network. The detection of faults is based on the analysis of residuals. A simple neural network trained to maximize the output entropy can approximate the probability density function of a residual and in this way a representative threshold value can be obtained with a given significance level. A self-organizing training to adjust network parameters is used. "The proposed approach, tested on the deviation faults of hyperchaotic Rossler system, exhibited a good performance in the fault detection, showing that it is suitable even for the diagnosis of nonlinear systems. In the future work the approach will be extended to the fault isolation and in estimating the entity of the parameter variation.

References

1. Bandlerand, J.W., Salama, A.: Fault diagnosis of analog circuits. In: Proc. IEEE ISCAS, pp. 1279–1325 (1985)
2. Abderrahman, A., Cerny, E., Kaminska, B.: Optimization Based Multifrequency Test Generation for Analog Circuits. Journal of Electronic Testing: Theory and Applications 9(1-2), 59–73 (1996)
3. Cherubal, S., Chatterjee, A.: Test Generation Based Diagnosis of Device Parameters for Analog Circuits. In: Proc. Design, Automation and Test in Europe Conf., pp. 596–602 (2001)
4. Cherubal, S., Chatterjee, A.: Test Generation Based Diagnosis of Device Parameters for Analog Circuits. In: Proc. Design, Automation and Test in Europe Conf., pp. 596–602 (2001)
5. Prasannamoorthy, V., Devarajan: Frequency domain technique for fault diagnosis in analog circuits-software and hardware implementation. Journal of Theoretical and Applied Information Technology 005 - 2010 JATIT & LLS
6. Zhou, L., Shi, Y., Zhao, G., Zhang, W., Tang, H., Su, L.: Soft-Fault Diagnosis of Analog Circuit with Tolerance Using Mathematical Programming. Journal of Communication and Computer 7(5) (Serial No.66) ISSN 1548-7709
7. Devarayanadurg, G., Soma, M.: Analytical Fault Modeling and Static Test Generation for Analog ICs. In: Proc. Int. Conf. on Computer-Aided Design, pp. 44–47 (November 1994)
8. Farchy, S.L., Gadzheva, E.D., Raykovska, L.H., Kouyoumdjiev, T.G.: Nullator-Norator Approach to Analogue Circuit Diagnosis Using General-Purpose Analysis Programmes. Int. Journal of Circuit Theory and Applications 23(6), 571–585 (1995)
9. Figueras, J.: Possibilities and Limitations of IDDQ Testing in Submicron CMOS. In: Proc. Innovative Systems in Silicon Conf., pp. 174–185 (October 1997)
10. Guo, Z., Savir, J.: Analog Circuit Test Using Transfer Function Coefficient Estimates. In: Proc. Int. Test Conf., pp. 1155–1163 (October 2003)
11. Halder, A., Bhattacharya, S., Chatterjee, A.: Automatic Multitone Alternate Test Generation for RF Circuits Using Behavioral Models. In: Proc. Intl. Test Conf., pp. 665–673 (November 2003)

12. Kondagunturi, R., Bradley, E., Maggard, K., Stroud, C.: Benchmark Circuits for Analog and Mixed-Signal Testing. In: Proc. 20th Int. Conf. on Microelectronics, pp. 217–220 (March 1999)
13. Panic, V., Milovanovic, D., Petkovic, P., Litovski, V.: Fault Location in Passive Analog RC Circuits by measuring Impulse Response. In: Proc. 20th Int. Conf. on Microelectronics, pp. 12–14 (September 1995)
14. Rajsuman, R.: IDDQ Testing for CMOS VLSI. Proceedings of the IEEE 88(4) (April 2000)
15. Slamani, M., Kaminska, B.: Analog Circuit Fault Diagnosis Based on Sensitivity Computation and Functional Testing. IEEE Design & Test of Computers 19(1), 30–39 (1992)
16. Frank, P.M., Köppen-Seliger, B.: New developments using AI in fault diagnosis. Artificial Intelligence 10(1), 3–14 (1997)
17. Gertler, J.: Fault detection and diagnosis in engineering systems. Marcel Dekker, New York (1998)
18. Schagaev, I.: Reliability of malfunction tolerance. In: Proc. of Int. Multiconference on Computer Science and Information Technologies, Wisla, Poland (October 2008)
19. Azam, M., Ioannides, N., Rümmeli, M.H., Schagaev, I.: System Software Support for Router Reliability. In: 30th IFAC Workshop on Real-Time Programming, and 4th International Workshop on Real-Time Software (WRTP/RTS 2009), Mragowo, Poland (October 2009)
20. Schagaev, I., Zalewski, J.: Redudancy classification for fault tolerant computer design. In: Proc. of the 2001 IEEE Systems, Man, and Cybernetics Conference, Tucson, Arizona (October 2001)
21. Yuan, L., He, Y., Huang, J., Sun, Y.: A New Neural-Network-Based Fault Diagnosis Approach for Analog Circuits by Using Kurtosis and Entropy as a Preprocessor. IEEE Transaction on Instrumentation and Measurement 59(3) (March 2010)
22. Back, A.D., Tsoi, A.C.: FIR and IIR Synapses, a New Neural Network Architecture for Time Series Modeling. Neural Computation 3, 375–385 (1991)
23. Spall, J.C.: Multivariate stochastic approximation using a simulta- neous perturbation gradient approximation. IEEE Trans. Autom. Control. 37(3), 332–341 (1992)
24. Bell, A.J., Sejnowski, T.J.: An information-maximization approach to blind separation and blind deconvolution. Neural Computation 7, 1129–1159 (1995)
25. Rössler, O.E.: An equation for hyperchaos. Physics Letters 71, 155–157 (1979)
26. Principe, J.C., Euliano, N.R., Curt Lefebvre, W.: Neural and Adaptive Systems. Wiley, Chichester (2000)
27. Bishop, C.M.: Neural Networks for Pattern Recognition. Clarendon Press, Oxford (1995)
28. Cybenko, G.: Approximation by superposition of a sigmoidal function. Mathematics of Control, Signal and Systems 2, 304 (1989)
29. The Mathworks, Neural Networks Toolbox for use with Matlab, v. 7.4.0.287, R (2007)