Coefficient-Based Parametric Faults Detection in Analog Circuits

Zhen Guo
Department of Electrical and Computer Engineering
New Jersey Institute Of Technology
Newark, NJ 07102, USA
zxg1554@njit.edu

ABSTRACT

Coefficient-based method is introduced for the parametric faults detection in analog circuits. By use of pseudo Monte-Carlo simulation we can greatly speed up the calculation of bounds of CUT transfer function's coefficients. We can estimate transfer function's actual numeric coefficients with system identification method. There is no need for a apriori knowledge of the symbolic transfer function. Finally we show that it is possible to determine whether any given CUT is faulty.

Categories and Subject Descriptors

B.7.3 [**Hardware**]: Integrated Circuits—*Reliability, Testing*

General Terms

Measurement

Keywords

Fault detection, Parametric faults, Monte-Carlo simulation, System identification

1. INTRODUCTION

The compactness of analog integrated circuits presents serious problems of accessibility. This makes analog testing very difficult. With the assumption of no internal node-probing, fault detection which is based on the prediction from the measurements on circuit under test (CUT) may be one of the solutions.

In this paper, we propose a coefficient-based method in which we use system identification method to predict the coefficients of CUT's transfer function. With the comparison between the estimated coefficient and its pre-defined bounds, we can detect if there is a parameteric fault in CUT.

Transfer function is widely used as a vehicle in analog testing [2],[5],[7], [8]. Let $H(s)$ be the transfer function [3] of the CUT. Since the CUT is linear and time-invariant, its transfer function can be expressed as

$$H(s) = K\frac{s^m + \sum_{i=0}^{m-1} a_i s^i}{s^n + \sum_{i=0}^{n-1} b_i s^i}, \quad n \geq m \tag{1}$$

The coefficients of the transfer function, i.e. K, a_i and b_i depend upon the circuit parameters p_i. As a result, the value of the coefficients of the good-machine are enclosed within their individual hyper-cubes. Whenever one or more of these coefficient values slip outside their hyper-cube we get a different transfer function that reflects the existence of a detectable fault [9].

This paper is organized as follows: In Section II, we propose a pseudo-Monte Carlo simulation method to speed up the calculation of the bounds of transfer function's coefficients with the supports of lemma and theorems. In Section III, we present the steps of how to estimate the numeric transfer function's coefficients. In Section IV, we elaborate on how to use coefficient-based method to detect faults. Conclusion is in Section V.

2. WHEN SYMBOLIC EXPRESSIONS ARE UNATTAINABLE

How can we determine if a circuit is faulty or fault-free when the symbolic transfer function is unattainable? This is likely to be the case when large analog circuits are being analyzed. In this case system identification method may be utilized to solve the problem.

Given that the netlist of the CUT is known (i.e. the model of the fault-free circuit), a nominal *numeric* transfer function may be calculated. The LNAPTF(Linear Networks Analysis Program for Transfer Function) software [4] may be used to obtain this numeric transfer function. Furthermore, the *numeric* upper and lower bounds of all transfer function coefficients may be obtained from a Monte-Carlo simulation. To obtain these numeric upper and lower bounds, the LNAPTF software is repeatedly invoked, the circuit parameters values are randomly chosen from their fault-free range ($[p_{in}(1-\alpha), p_{in}(1+\alpha)]$, for all i). Upon completion of this Monte-Carlo simulation, all the numeric values of $a_{i,min}$ and $a_{i,max}$ and $b_{j,min}$ and $b_{j,max}$ (see Eq. 1) are known to some degree of confidence[1]. In order to speed up the Monte Carlo simulation, we proposed the following lemma and theorems.

[1]Determined by the number of iterations of the Monte-Carlo simulation process

GLSVLSI'03, April 28–29,2003, Washington DC,USA

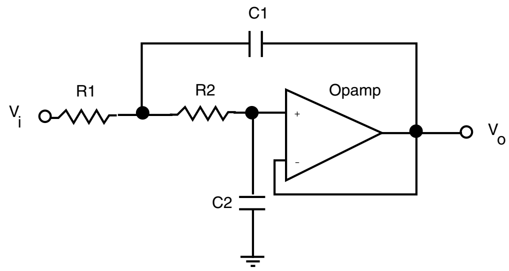

Figure 1: A low-pass filter

Lemma 1: *Any coefficient of the transfer function of analog linear circuits can be written in the format of $\frac{N}{D}$,where N and D are in the format of SOP(sum of products) of circuit parameters.*

Example 1: Consider the low-pass filter of Fig.1. The circuit parameters are $R_1 = R_2 = 2.26k\Omega, C_1 = 0.02\mu F, C_2 = 0.01\mu F$. The circuit transfer function, assuming the operational amplifier has an infinite gain, is given by:

$$H(s) = \frac{V_o(s)}{V_i(s)} = \frac{\frac{1}{R_1R_2C_1C_2}}{s^2 + \frac{R_1+R_2}{C_1R_1R_2}s + \frac{1}{R_1R_2C_1C_2}}$$

We show this example to verify *Lemma 1.*

Theorem 1. *In the coefficients of the transfer function of any analog linear circuit, no parameter (circuit parameter) can have a degree greater than one.*

Theorem 2. *The minimum and maximum of the transfer function's coefficients $y = y(X_1, X_2, ..., X_n)$ occur on the extreme ends of the parameter values (any parameter's degree is no more than one)*

Remarks: With *Lemma 1*, *Theorem 1* and *Theorem 2*, we know any coefficient's bounds will occur on the boundary value of each circuit parameter. Therefore we only need feed the combination of boundary values of each parameter into LNAPTF instead of randomly choosing the circuit parameters from their fault-free range ($[p_{in}(1-\alpha), p_{in}(1+\alpha)]$, for all i). Therefore we call our Monte-Carlo simulation as pseudo Monte-Carlo simulation.

3. HOW TO ESTIMATE TRANSFER FUNCTION

3.1 Selection of input signal

In order to compute the actual (numeric) transfer function of the CUT, we excite the circuit with a frequency-rich input. In parameter estimation, we need the input signal to have a big frequency range. Suppose we do not know the specification of the system before parameter estimation and if we select a fixed frequency by random which is not suitable to the system, there can be a negative effect on the parameter estimation result. In our experiment, we use pulse as our input signal.

3.2 Noice estimation

If systems were deterministic, in the sense that all the inputs could be specified exactly and all the outputs could be measured with an unlimited precision. In [3], "If no noise exists in measurements, then the identification problem is not difficult, however, in practice, noise often arises in measurements.", so it is important to estimate the measurment noise before the parameter estimation. Normally measurement noise is simulated by adding Gaussian distributed random numbers to the simulated test response.

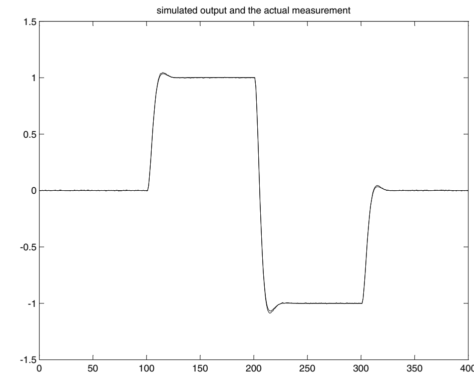

Figure 2: Lowpass filter: Simulated outputs and true measurements

3.3 Estimating the actual transfer function

The circuit response to the pulse input is measured and sampled over a given period of time. The input and the (measured) sampled outputs are inputed to a system identification tool, such as the one existing inside MATLAB [1]. We have used an ARX-based code to assist in computing the estimated transfer function.

The ARX [6] model predicts the next output $\hat{y}(t)$ from the previous measured data of the output $(y(t-1), y(t-2), ..., y(t-k))$ and the input data $(u(t-1), u(t-2), ...u(t-k))$. The output error between the predicted output calculated from the ARX model and the measured output from the real system always exists. Fitting the model to the system is in fact a minimization problem of this error by selecting the model parameters. LSE(Least Square Estimation) is used in the the minimization of this error. Finally we obtain the estimated parameters which lead to the minimization of the error.

Example 2: Conside second-order low-pass filter of Fig. 1. The true transfer function is $\frac{9.789e008}{s^2+4.425e004s+9.789e008}$ and the estimated transfer function is $\frac{-177.4s+1.026e009}{s^2+4.697e004s+1.025e009}$

In continuous time domain, the distance of a pole or zero from the imaginary axis determine the effect on the overall system response. In this case, compared with poles, zeros are much far away from imaginary axis, therefore it can be ignored as: the estimated transfer has a zero at 1.026e009/177.4, which is very far from the imaginary axis , therefore can be ignored: -177.4s + 1.026e009 $\cong 1.026e009$

The simulation results show the estimated time response (Fig.2) and frequency response(Fig. 3) are very close to the actual ones.

Notice this is a typical situation when system identification method is used to estimate analog linear circuits. This identification procedure, however, introduces new coefficients which do not exist in $H_n(s)$. The reason is when we use ARX model, we need to know the order of the circuit. Usually a circuit's order means the order of the denominator

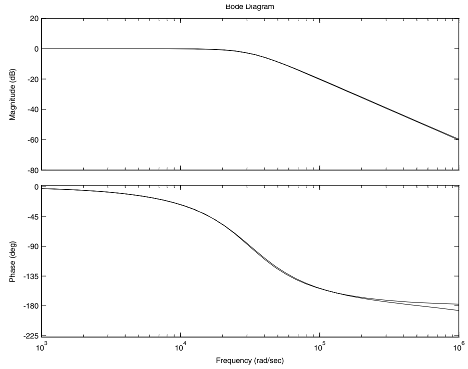

Figure 3: Comparsion of frequency and phase responses

of its transfer function. When we use ARX model, we need to give the order of the denominator as well as the numerator's. In our example above, we assume we do not know the exact order of the numerator, so we simply give the same order to the numerator as the denominator. That is why in the above case, one new coefficient was introduced in the numerator.

Remarks: For the case of parametric faults, we are able to assume that we know the exact structure(especially the orders of the numerator) of the analog circuits. The reason is that parametric faults referring to small changes in a circuit usually do not affect the circuit's connectivity. Parametric faults will not increase the order of the numerator and denominator. Let's consider two special conditions. One is the missing term condition, which means one or more coefficients missing because of the parametric faults. In order to use system identification methods, we regard the missed coefficients as they change to zero and we can pretend they are still there. The other condition is one or more coefficients changing to infinity. This is a complex case.

Here is a second-order system

$$H(s) = \frac{ds + e}{as^2 + bs + c} \tag{2}$$

(1)If a changes to infinity, the poles of the system approach to the orginal point, the output will be zero

(2)If b changes to infinity, the poles of the system approach to the infinity also, the output will be zero.

(3)If c changes to infinity, the poles of the system approach to the infinity also, the output will be zero.

(4)If d changes to infinity, the zeros of the system approach to the original point, the output will be infinity.

(5)If e changes to infinity, the zeros of the system approach to the infinity, the output will be infinity.

From this example, we can see that if one of the coefficients changes to infinity, the output will be either infinity or zero. For the infinity case, it is out of the scope of any measurement devices and it will result in device saturation. System identification method can only handle bounded inputs and bounded outputs. It can not handle the cases when the output is infinity or zero. So when we detect parametric faults with system identification methods, we can preclude the coefficient's "infinity" case. Therefore when we post-process the estimated "raw" transfer function of CUT, we can simply match the coefficients in the estimated "*raw*"

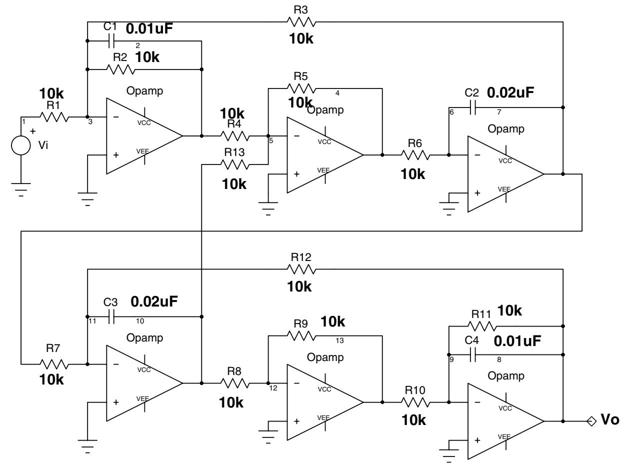

Figure 4: Leapforg filter

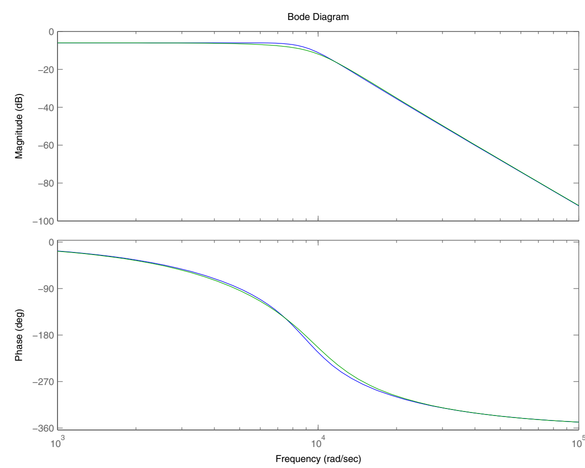

Figure 5: Leapfrog filter: comparison of frequency and phase between estimated one (after post-processing) and nominal one

numerator with the coefficients in the nominal numerator. The extra coefficients which do not exist in the nominal numerator will be ignored.

Example 3: Conside a 4th-order leapfrog filter of Fig. 4. The true transfer function is $\frac{2.5e15}{s^4+20000s^3+2.25e08s^2+1.5e012s+5e15}$ and the estimated "raw" transfer function is $\frac{0.0008638s^4+174.9s^3-6.117e06s^2-8.181e10s+2.545e15}{s^4+1.986e04s^3+2.429e08s^2+1.573e12s+5.102e15}$

After matching the coefficients of the "raw" transfer function with the nominal transfer function, only $2.545e15$ is left in the numerator and all others are ignored. Finally we obtain $\frac{2.545e15}{s^4+1.986e04s^3+2.429e08s^2+1.573e12s+5.102e15}$

The comparison of frequency and phase between the estimated one (after coefficient matching) and the nominal one is shown in Fig. 5. The result is good enough for identification purpose. This example indicates that we can use the coefficient matching method to speed up the post-processing of the estimated "raw" transfer function.

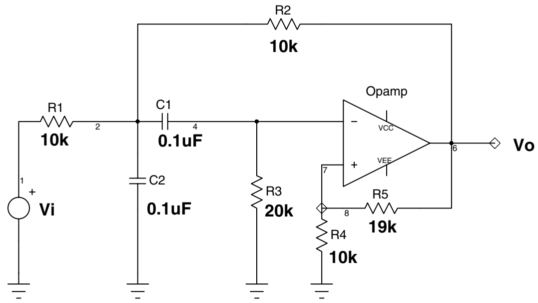

Figure 6: Second order bandpass filter

Table 1: Parameter Combinations leading to the coefficients' bounds

Coeff	R1(Ω)	R2(Ω)	R3(Ω)	R4(Ω)	R5(Ω)	C1(F)	C2(F)
K_{min}	10.5K	9.5K	19K	10.5K	18.05K	0.95e-7	1.05e-7
K_{max}	9.5K	9.5K	19K	9.5K	19.95K	0.95e-7	0.95e-7
$b_{1,min}$	10.5K	9.5K	21K	9.5K	19.95K	1.05e-7	0.95e-7
$b_{1,max}$	9.5K	10.5K	19K	10.5K	18.05K	0.95e-7	1.05e-7
$b_{2,min}$	9.5K	9.5K	19K	9.5K	18.05K	0.95e-7	0.95e-7
$b_{2,max}$	10.5K	10.5K	21K	9.5K	18.05K	1.05e-7	1.05e-7

4. HOW TO DETECT FAULTS

All the information is now available for determining whether or not the CUT is faulty. Once these actual coefficient values are known, we proceed to examine whether any one of them slips outside its prescribed fault-free range (computed earlier). If one, or more, of these coefficients is found to be outside their fault-free bounds, the CUT is declared faulty;

Consider the band-pass filter of Fig. 6. Even though this circuit is small, we will pretend not to know its symbolic transfer function. We only use the knowledge that it is a second order circuit. The nominal numeric transfer function for this circuit is calculated by LNAPTF from its netlist information.

$$H_n(s) = \frac{2900s}{s^2 + 100s + 10^6} \tag{3}$$

Choosing each parameter's normal drift $\alpha = 0.05$, the Monte-Carlo simulation yields the following coefficients' bounds and also we verified the Theorem 2 that the each coefficient's bounds occur on its boundary values (see Tab. I):

$$K_{min} = 2627, \qquad K_{max} = 3083 \tag{4}$$

$$b_{1,min} = -200.75, \qquad b_{1,max} = 377.1 \tag{5}$$

$$b_{2,min} = 8.885 \times 10^5, \qquad b_{2,max} = 1.13 \times 10^6 \tag{6}$$

Notice if the system is stable, then the coefficient b1 should be positive. Otherwise the system is unstable. So we need to modify the bound of the coefficient b1 to $[0, 377.1]$.

The fault C1=.125μF (25% off on the up-side) has been injected into the circuit. The CUT's transfer function is estimated as:

$$H(s) = \frac{2893s - 67.28}{s^2 + 9.437s + 7.684 \times 10^5} \tag{7}$$

After post-processing with coefficient-matching of the Eq. 7, we obtain $\frac{2893s}{s^2+9.437s+7.684\times10^5}$. When we compare all the estimated coefficients against their fault-free bounds, we discover that b_2 is outside its fault-free range. The conclusion is that the CUT is faulty. There are some other detection examples shown in the Tab. II.

Table 2:

Parameters	Estimated transfer function	Coefficients' status	Detect
R1 25% up	$\frac{2320s}{s^2+11.784s+9e05}$	K out of bound	✓
R2 20% up	$\frac{2922s}{s^2+580.3s+9.267e05}$	b_1 out of bound	✓
R3 20% up	$\frac{2893s}{s^2+66.67s+8.333e05}$	b_2 out of bound	✓
R4 10% down	$\frac{2982s}{s^2-10.51s+1e06}$	b_1 out of bound	✓
R5 25% down	$\frac{27663s}{s^2+436.7s+1.009e06}$	b_1 out of bound	✓
C1 25% up	$\frac{2893s}{s^2+9.437s+7.684e05}$	b_2 out of bound	✓
C2 15% down	$\frac{3408s}{s^2+37.14s+1.097e06}$	K out of bound	✓

5. CONCLUSIONS

We have shown that by the use of pseudo Monte-Carlo simulation and system identification method it is possible to determine whether any given CUT is faulty. There is no need for a apriori knowledge of the symbolic transfer function.

System identification method's being applied in the fault detection of analog circuits is a new topic, which is different from its application in automatic control domain. One difference is whether we can assume that we know the exact structure(the orders of numerator and denominator) of the analog circuits or not. For the purpose of detecting parametric faults, the assumption stands.

6. REFERENCES

[1] *MATLAB Version 6 System Identification toolbox.*

[2] A. Abderrahman, E. Cerny, and B. Kaminska. Worst case tolerance analysis and clp-based multifreqyuency test generation for analog circuit. *IEEE Trans. On Computer-Aided Design*, 18(3):332–345, 1999.

[3] Chi-Tsong Chen. *Analog and Digital Control System Design: Transfer Function, State-Space, and Algebraic Methods.* 1993. Saunders College Publishing.

[4] P. Clayton. *Analysis of Linear circuits.* 1989. McGraw-Hill.

[5] G. J. Hemink. Testability analysis of analog systems. *IEEE Trans. On Computer-Aided Design*, 9(6):573–583, 1999.

[6] Lennart Ljung. *System Identification: Theory for the User.* 1987. Prentice-Hall,NJ.

[7] R. Ramadoss and Michael L. Bushnell. Test generation for mixed signal devices using signal flow graph. *Journal of Electronic Testing: Theory and Application*, 14(3):189–205, 1999.

[8] J. Savir and Z. Guo. On the dectability of parametric faults in analog circuits. In *Proc. ICCD*, pages 273–276, September 2002.

[9] J. Savir and Z. Guo. Test limitation of parametric faults in analog circuits. In *Proc. ATS'02*, pages 39–44, November 2002.