POLYPUBLIE

Génération de tests robustes pour les circuits analogiques linéaires

Abdessatar Abderrahman

Document en libre accès dans PolyPublie
Open Access document in PolyPublie

Date	1997
Type	Mémoire ou thèse / Dissertation or Thesis Thèse de doctorat/Ph.D. thesis
URL de PolyPublie PolyPublie URL	https://publications.polymtl.ca/7062/
Conditions d'utilisation Terms of use	© Tous droits réservés / All rights reserved
Référence Citation	Abderrahman, A. (1997). Génération de tests robustes pour les circuits analogiques linéaires [Thèse de doctorat, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/7062/

Détails de la thèse
Thesis details

Directeurs de recherche Advisors	Bozena Kaminska
Programme Program	Non spécifié

Ce fichier a été téléchargé à partir de PolyPublie, le dépôt institutionnel de Polytechnique Montréal
This file has been downloaded from PolyPublie, the institutional repository of Polytechnique Montréal
https://publications.polymtl.ca

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

UMI®

UNIVERSITÉ DE MONTRÉAL

GÉNÉRATION DE TESTS ROBUSTES POUR LES CIRCUITS ANALOGIQUES LINÉAIRES

par

Abdessatar ABDERRAHMAN

DÉPARTEMENT DE GÉNIE ÉLECTRIQUE ET DE GÉNIE INFORMATIQUE

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

THÈSE PRÉSENTÉE EN VUE DE L'OBTENTION

DU DIPLÔME DE PHILOSOPHIAE DOCTOR (Ph.D)

(GÉNIE ÉLECTRIQUE)

OCTOBRE 1997

© Abdessatar Abderrahman 1997.

National Library of Canada
Acquisitions and Bibliographic Services
395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale du Canada
Acquisitions et services bibliographiques
395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-73437-4

Canada

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Cette thèse intitulée:

GÉNÉRATION DE TESTS ROBUSTES

POUR LES CIRCUITS ANALOGIQUES

LINÉAIRES

présentée par: ABDERRAHMAN Abdessatar

en vue de l'obtention du diplôme de: Philosophiae Doctor

a été dûment acceptée par le jury d'examen constitué de:

M. SAVARIA Yvon, Ph.D., président

Mme KAMINSKA Bozena, Ph.D., membre et directrice de recherche

M. CERNY Eduard, Ph.D., membre et codirecteur de recherche

M. ABOULHAMID Mostapha, Ph.D., membre

M. THIBEAULT Claude, Ph.D., membre

DÉDICACE

A l'âme de mon père, à ma mère,

à ma femme et à mes enfants.

Remerciements

Je désire, en premier lieu, témoigner ma reconnaissance à ma directrice de recherche Madame Bozena Kaminska, professeur à l'École Polytechnique de Montréal, et à mon codirecteur Monsieur Eduard Cerny, professeur à l'Université de Montréal, dont la supervision a permis de mettre en oeuvre ce travail. Je remercie ma directrice pour ses conseils précieux, ses encouragements et son support financier, ainsi que la confiance qu'elle a mise en moi pour accomplir ce travail. Je remercie mon codirecteur pour ses sages directives, l'encadrement fructueux dont j'ai bénéficié, son support financier, ainsi que ses qualités professionnelles et humaines que j'ai admirées.

J'aimerais remercier l'organisme "Groupe Inter-Universitaire en Architecture des Ordinateurs et VLSI (GRIAO)" pour le support financier dont j'ai bénéficié durant l'accomplissement de cette thèse.

Je tiens à remercier tout particulièrement ma mère, Fatma, pour ses sacrifices sans limites, ma femme, Sayda, pour son appui persévérant, et mes filles Maroua, Mona, Safa et Ela pour le temps que je leur dois.

Je voudrais remercier Monsieur Yvon Savaria d'avoir accepté de présider la commission de jury. J'aimerais également remercier Monsieur Mostapha Aboulhamid et Monsieur Claude Thibeault d'avoir accepté de faire partie de cette commission. Je remercie aussi Monsieur Guy Bois d'avoir fait partie du jury de mon examen général de synthèse.

RÉSUMÉ

Les circuits analogiques sont traditionnellement testés par la vérification de leur fonctionnalité. Or, ce genre de test est reconnu très coûteux. En effet, le coût de test analogique peut représenter jusqu'à 30% du coût total de fabrication. De plus, dans certaines applications, une proportion de 95% du coût total de test est dépensée dans le test analogique et 5% seulement dans le test de la partie numérique, alors que la puce est constituée dans une proportion de 95% de circuits numériques et 5% seulement de circuits analogiques. En outre, dans ce type de test les circuits sont sur-testés ou sous-testés puisqu'il n'y a pas de critère d'arrêt. Il en résulte, qu'afin de minimiser le coût de test de production, il est nécessaire de développer des techniques de génération de tests basées sur des modèles de défaut. En général, on distingue dans les circuits analogiques deux catégories de défauts: défauts paramétriques et défauts catastrophiques.

Dans cette thèse nous proposons une méthode de génération de tests basée sur ces modèles de défaut. Cette méthode vise à générer un ensemble minimal de tests robustes permettant d'assurer pour chacun des composants (paramètres) du circuit sous test une observabilité maximale de pannes. Cet ensemble de tests permettra de distinguer les circuits défectueux de ceux qui sont bons. Un ensemble de tests est dit robuste s'il est en mesure de détecter les défauts paramétriques et catastrophiques sous la contrainte d'effet maximal de masquage dû à la tolérance des composants du circuit. En effet, seulement dans ce cas, la qualité d'un test peut être correctement mesurée et garantie. La génération

d'un ensemble de tests ayant de telles spécifications en particulier, et le test des circuits analogiques en général, sont des problèmes difficiles. Cette difficulté émane de la complexité naturelle de ces circuits qui présentent un spectre continu de défauts. Cela provient de la continuité des variations dans le temps des grandeurs physiques qui caractérisent les circuits analogiques. Les effets de masquage causés par les tolérances des paramètres ainsi que l'incertitude que ces derniers induisent sur les caractéristiques de ces circuits compliquent encore plus le problème du test analogique. En outre, la non-linéarité de ces caractéristiques ainsi que le manque d'accessibilité aux noeuds internes rendent ce problème plus complexe.

Nous avons élaboré dans cette thèse une méthode de génération de tests multifréquences pour détecter les pannes paramétriques et catastrophiques dans les circuits analogiques linéaires. L'originalité de cette méthode réside dans le fait que le problème de génération de tests est formulé comme une série de problèmes d'optimisation. Étant donné un ensemble de performances (spécifications du circuit sous-test) et une plage de fréquences, la méthode proposée sélectionne les fréquences de tests qui maximisent l'observabilité des déviations d'un paramètre défectueux du circuit, observées à travers une performance, sous la contrainte de masquage maximal induit par les autres paramètres.

Initialement, nous avons résolu les problèmes d'optimisation formulés à l'aide d'une méthode de programmation non-linéaire, en l'occurrence la programmation quad-

ratique séquentielle (SQP) disponible dans l'outil de mathématiques MATLAB. Cependant de telles méthodes d'optimisation ne peuvent pas garantir que la solution trouvée est globale. Cela peut donc conduire à une sélection erronée de tests affectant ainsi la qualité de l'ensemble de tests générés. De plus, ces méthodes ne sont pas automatiques et dépendent de plusieurs paramètres qui doivent être choisis par un usager expérimenté. Nous avons alors élaboré une nouvelle méthode basée sur la programmation logique à contraintes, qui permet de résoudre les problèmes d'optimisation formulés comme une série de problèmes de satisfaction à contraintes. Cette méthode est automatique et génère des bornes étroites, infaillibles et garanties à l'intérieur desquelles se trouve les optimums (minimum et maximum) globaux d'une fonction non-linéaire. Cette qualité d'étroitesse et d'infaillibilité de ces bornes trouve son origine dans les propriétés de l'arithmétique relationnelle des intervalles sur laquelle est basée la programmation logique à contraintes (CLP) qu'on a utilisée pour réaliser ce travail. Parmi ces propriétés figurent la propriété d'inclusion des intervalles, le rétrécissement systématique de ces derniers à travers la propagation des contraintes, l'exactitude de ce processus, ainsi que l'arrondissement vers l'extérieur des résultats des calculs.

Nous illustrons l'efficacité de la méthode proposée sur des fonctions mathématiques non-linéaires reconnues difficiles. Puis, nous l'appliquons à des circuits électriques d'essai dans le contexte de la génération de tests.

Cette méthode est aussi utilisée pour résoudre le problème d'analyse de tolérance des performances d'un circuit dans le pire des cas. Cette analyse de tolérance joue un rôle important dans le domaine de vérification de design. Ce problème est efficacement traité ici comme une composante du problème de génération de tests. La méthode proposée est également utilisée pour analyser la testabilité des circuits analogiques linéaires, en vue d'augmenter l'observabilité des pannes grâce à la technique de design orienté testabilité (DFT).

Dans le premier chapitre nous mettrons en évidence l'importance du test analogique dans la mise en oeuvre des produits électroniques modernes, et la nécessité de mise au point de nouvelles méthodes de génération de tests plus performantes et plus économiques. Le sujet ainsi que les objectifs de cette thèse sont également définis dans le chapitre 1. Dans le chapitre 2, nous situerons ce sujet par rapport à la discipline dans laquelle il s'inscrit, et nous passerons en revue différents travaux accomplis dans le domaine du test analogique. L'analyse et la formulation du problème de génération de tests, ainsi que certains résultats expérimentaux et leur validation seront présentés dans le chapitre 3. La méthode d'optimisation globale proposée pour résoudre le problème de génération de tests et celui d'analyse de tolérance est décrite dans le chapitre 4. Le chapitre 5 est consacré à l'analyse et l'amélioration de la détectabilité des pannes dans les circuits analogiques linéaires. Enfin, nous concluons dans le chapitre 6.

Abstract

Analog circuits are traditionally tested by verifying their function, which is known to be costly. Indeed, the estimated cost of analog testing may represent 30% of the total manufacturing cost. Furthermore, in some applications, 95% of total testing costs are spent in testing of analog circuits and 5% in testing of digital circuits, while 95% of the chip are digital and 5% are analog. Moreover, in this type of analog testing there is a lack of quantitative measure of a test quality. As a result, the circuits are over-tested or under-tested. Consequently, to minimize the test time and thus the cost of production testing of analog circuits, test generation techniques based on fault-models are required. In general, faults in analog circuits can be classified into catastrophic and parametric faults. A robust test set for analog circuits has to detect faults under maximal masking effects due to variations of circuit parameters in their tolerance box. Indeed, only in this case the quality of a test set may be correctly measured and guaranteed. In this thesis we elaborated a multifrequency test generation method (TPG) for detecting parametric and catastrophic faults in linear analog circuits.The originality of this method stems from the formulation of analog test generation problem as a series of optimization problems. Given a set of performances and a frequency range, our approach selects the test frequencies that maximize the observability of a parameter deviation on a circuit performance under the worst masking effects of normal variations of the other parameters. Initially, we solved these optimization problems by Sequential Quadratic Programming (SQP), a non-linear programming method available in MATLAB. Such optimization

methods use local information and consequently cannot guarantee a global optimum. Furthermore, they are not automatic and depend on various parameters which must be selected by an experienced user. We thus developed a new method based on Constraint Logic Programming(CLP) that solves the optimization problems in TPG as a series of Constraint Satisfaction Problems (CSPs). The TPG method is fully automatic and provides tight and guaranteed bounds on the global optima of a nonlinear function. This quality of tightness and correctness of the bounds as assured by our technique stems from the inclusion properties of intervals, outward rounding, and the correctness of narrowing on which is based CLP implemented using the Relational Interval Arithmetic (RIA). The TPG method was implemented in CLP(BNR) Prolog. We first illustrate the effectiveness of our approach on a number of nonlinear functions known to be difficult, and then we apply it to a realistic electronic circuit in the context of TPG.

Our method may also be used to solve worst-case tolerance analysis of circuit performances problem which plays an important role in design verification. This problem is efficiently tackled here as part of the test generation. Moreover, our approach is useful to analyze the testability of linear analog circuits, and thus permits to enhance the fault detectability thanks to design for testability.

The importance of analog testing and the need to develop new efficient and less costly methods are shown in Chapter 1. The objectives of the thesis are also defined in this chapter. Some important results on analog testing are reviewed in Chapter 2. Analy-

sis and formulation of the test generation problem are developed in Chapter 3. The test generation algorithm, some experimental results, and the test set validation are also described in Chapter 3. The new global optimization method based on CLP for solving the test generation and worst-case tolerance analysis problems are outlined in Chapter 4. The application of the proposed test generation method to analyze the testability of linear analog circuits is illustrated in Chapter 5. Finally, some conclusions are drawn in Chapter 6.

TABLE DES MATIÈRES

DÉDICACE iv

REMERCIEMENTS v

RÉSUMÉ vi

ABSTRACT x

TABLE DES MATIÈRES xiii

LISTE DES FIGURES xx

LISTE DES TABLEAUX xxii

Chapitre 1 -Introduction 1

1.1 Objectifs de la thèse 4

1.2 Analyse du problème 5

Chapitre 2 -Revue de littérature 9

2.1 Concepts de base 9

2.1.1 Test et diagnostic 9

2.1.2 Défauts physiques 9

2.1.3 Modelisation 11

2.1.4 Simulation 12

2.1.5 Modèle de défaut 12

2.1.6 Evaluation d'un test et simulation de pannes 15

2.2 Méthodes de test 16

2.2.1 Méthode de test en temps réel (On-line testing) 16

2.2.2 Méthode d'auto-contrôle de design (Self-checking design) 17

2.2.3 Méthode d'auto-test incorporé (Built_In Self Testing) 17

2.2.4 Méthode de test à sonde guidée 18

2.2.5 Méthode de test en circuit (In_circuit testing) 19

2.2.6 Méthode de test par comparaison 19

2.2.7 Méthode de test par compression de la réponse 19

2.3 Modèle de défaut et simulation de pannes 19

2.4 Design orienté testabilité (Design For Testability: DFT) 22

2.4.1 Approche ad hoc 24

2.4.2 Approches stucturées: Techniques de balayage) 25

2.5 Diagnostic 29

2.6 Génération de tests 31

Chapitre 3 -Algorithme de génération de tests robustes pour les circuits analogiques linéaires 40

3.1 Introduction 43

3.2 The proposed approach.....45

3.2.1 Objectives 45

3.2.2 Problem analysis 46

3.2.3 Problem formulation 47

3.2.3.1 Valid range determination 47

3.2.3.2 Minimum observable parameter deviatio 48n

3.3 Test generation algorithm 51

3.4 Experimental Results 54

3.4.1 An illustrative example 54

3.4.2 A Realistic Application 58

3.4.3 Test set validation 62

3.5 Conclusions 65

Chapitre 4 -Algorithmes de recherche de solution globale des problèmes de génération de tests et d'analyse de tolérance, basés sur la technique CLP 73

4.1 Introduction 77

4.2 Problem formulation 81

4.2.1 Objectives 81

4.2.2 Problem Formulation 81

4.2.2.1 Worst-case tolerance analysis 82

4.2.2.2 Minimum observable parameter deviation [12] 84

4.3 Constraint Satisfaction Problems 86

4.3.1 Constraint Logic Programming 86

4.3.1.1 Functional Interval Arithmetic 86

4.3.1.2 Relational Interval Arithmetic (RIA) 88

4.3.1.3 Constraint Satisfaction Problems 89

4.4 Proposed CLP-based approach for solving optimization problems 91

4.4.1 Worst-case tolerance analysis 91

4.4.1.1 The basic approach 92

4.4.1.2 Description of the Optimization Algorithm 94

4.4.1.3 Numerical Examples 97

4.4.2 Finding optima (max and min) of observable parameter values 99

4.5 Experimental Results 103

4.5.1 Example 1 103

4.5.2 Finding optima (max and min) of observable parameter values 99

4.6 Experimental Results 103

4.6.1 Example 1 103

4.6.2 Example 2 106

4.6.3 Test quality 108

4.6.3.1 Effects of tester resolution 109

4.6.3.2 Effects of parameter tolerances 110

4.7 Acceleration techniques 112

4.7.1 Monotonicity test 112

4.7.1.1 Computation of the worst-case circuit response 112

4.7.1.2 Computation of observable parameter values 113

4.7.2 Mean-value form 114

4.7.3 Reducing search space of parameters 116

4.8 Conclusions 117

Chapitre 5 -Analyse et amélioration de la détectabilité des pannes. 123

5.1 Introduction 123

5.2 Définitions 125

5.3 Analyse de détectabilité du filtre biquadratique 128

5.3.1 Observation des gains V3 et V5 128

5.3.2 Observation des phases aux noeuds 3 et 5 130

5.3.3 Observation du facteur Q 130

5.3.4 Application de la technique DFT 131

5.4 Analyse de détectabilité d'un amplificateur à large bande 135

5.4.1 Amplificateurs opérationnels de gain infini 135

5.4.2 Amplificateurs opérationnels de gain fini 136

5.4.3 Amplis opérationnels de gain dépendant de la fréquence 137

5.4.4 Application des techniques DFT 139

5.5 Conclusion 142

Chapitre 6 -Conclusion Générale 143

BIBLIOGRAHIE 148

LISTE DES FIGURES

Figure 2.1:Chaîne de balayage analogique 26

Figure 3.1:Total possible range of a parameter x_m under normal variations of x_i 48

Figure 3.2:Low-pass filter 55

Figure 3.3:Extreme values T_{max}, T_{min} of pass-low filter gain 57

Figure 3.4:Observable minimum (maximum) deviation of parameter R_1 57

Figure 3.5:Observable minimum (maximum) deviation of parameter R2 58

Figure 3.6:Observable minimum (maximum) deviation of parameter C 58

Figure 3.7:biquadratic filter 60

Figure 3.8:Faulty response V5Ft observed under the smallest observable positive deviation (S.O.P.D) of R_8, compared to the envelope of the normal range (V5-min, V5-max) ... 66

Figure 3.9:Faulty response V5Ft observed under the largest observable negative deviation (L.O.N.D) of R_8, compared to the envelope of the normal range (V5-min, V5-max) ... 67

Figure 3.10:Faulty and good response families under faulty R_3 and the other parameters varied randomly 69

Figure 3.11:Faulty and good response families under faulty R_8 and the other parameters varied randomly 70

Figure 3.12:Faulty and good family responses under a faulty parameter R_1 and other random parameter variations 71

Figure 4.1: A biquadratic filter 105

Figure 4.2: A notch filter 108

Figure 4.3:Nominal and worst-case magnitude of notch filter 108

Figure 5.1:Filtre biquadratique 128

Figure 5.2:Amplificateur à large bande en cascade 136

LISTE DES TABLEAUX

TABLE 3.1 : Computed and simulated results 59

TABLE 3.2 : Computed results 62

TABLE 4.3 : Range of three-hump-camel function 98

TABLE 4.4 : Range of six-hump-camel function 99

TABLE 4.5 : Range of a composed function 100

TABLE 4.6 : Results for biquadratic filter 106

TABLE 4.7 : Results for notch filter 109

TABLE 4.8 : Tester resolution impact on parameter detectability 111

TABLE 4.9 : parameter tolerances impact on parameter detectability 112

TABLE 4.10 : Effect of monotonicity test 114

TABLE 4.11 : Speeding-up the computation of SOPD and LOND values of biquad parameters 117

TABLEAU 5.1 : Résultats obtenus en observant les gains V3 et V5 130

TABLEAU 5.2 : Résultats obtenus en observant les phases aux noeuds 3 et 5..... 131

TABLEAU 5.3 : Résultats obtenus en observant V3 133

TABLEAU 5.4 : Résultats obtenus pour des gains d'amplis infinis 137

TABLEAU 5.5 : Résultats obtenus pour des gains d'amplis op.finis 138

TABLEAU 5.6 : Résultats obtenus pour des gains d'amplis op. finis dépendant de la fréquence 140

TABLEAU 5.7 : Résultats obtenus avec le circuit modifié 142

Chapitre 1
Introduction

Depuis l'avènement des circuits intégrés, la densité d'intégration des transistors et les performances des circuits VLSI (Very Large Scale Integration) ne cessent de s'améliorer. Cette croissance rapide de la complexité des circuits VLSI, atteignant déjà quelques millions de transistors, et la prolifération sans précédent des circuits mixtes ont rendu le test de ces circuits une tâche ardue et très coûteuse. Alors que dans l'univers numérique plusieurs algorithmes et outils CAO de test sont de nos jours couramment utilisés dans l'industrie électronique, l'univers de test analogique accuse un certain retard. En effet, le test des circuits analogiques n'a été l'objet d'une attention accrue et intense que tout récemment. Cet intérêt si important trouve son origine dans l'augmentation considérable du nombre d'applications analogiques et mixtes. En fait, depuis l'avènement des circuits multipuces, il est devenu possible d'intégrer sur un même substrat des circuits analogiques et numériques et de les interconnecter. Par conséquent, depuis cet avènement on assiste à une prolifération sans précédent des circuits mixtes dans des nombreuses applications industrielles, plus particulièrement dans le traitement des signaux et des images. Cette nouvelle tendance a introduit de nouveaux défis dans le test des circuits mixtes provenant essentiellement de la difficulté de tester la partie analogique. Cette tâche est rendue plus compliquée par une accessibilité limitée aux noeuds internes de ces circuits et un nombre restreint de broches. En conséquence, le temps nécessaire de mise en marché des circuits analogiques

et mixtes devient dominé par le développement d'algorithmes et de programmes de test de la partie analogique.

L'un des problèmes importants dans le domaine du test est de pouvoir évaluer un ensemble de tests en termes d'efficacité et de qualité. Autrement dit, si l'on applique un ensemble de tests à un circuit, qu'est-ce qui nous permet de dire que ce circuit est correctement testé? L'évaluation d'un ensemble de tests se fait dans le cadre d'un modèle de défaut. Dans le cas des circuits numériques la qualité d'un ensemble de tests est mesurée par une quantité bien connue, à savoir la couverture des défauts. Celle-ci est souvent définie comme le pourcentage de défauts "collés à" détectés par l'ensemble de tests, divisé par le nombre total des défauts "collés à" qui peuvent se produire. La mesure de la qualité d'un ensemble de tests nous permet aussi de connaître les défauts non détectés et de modifier le circuit en conséquence pour augmenter sa testabilité (et de pratiquer ainsi du design orienté testabilité "DFT"). Par contre, dans l'univers analogique il est plus difficile d'évaluer et de garantir la qualité d'un test analogique. Cela provient du fait qu'il n'existe pas de modèle analogique universel et adéquat de défaut, tel que c'est établi dans l'univers numérique. Seules les défauts catastrophiques présentent une certaine analogie au modèle numérique "collé à". Dans ce cas, la simulation de défauts peut être utilisée pour quantifier la qualité d'un test en évaluant sa couverture de défauts. Il est aussi possible dans ce cas, de construire des dictionnaires de défauts pour le diagnostic des circuits analogiques. Il est clair, que le manque d'un modèle analogique de défaut approprié, plus particulièrement dans le cas de défauts paramétriques, constitue un handicap sérieux qui affecte le développement du test analogique.

Les circuits analogiques sont traditionnellement testés par la vérification de leur fonctionalité. Or, ce genre de test est reconnu très coûteux. En effet, le coût de test analogique peut représenter jusqu'à 30% du coût total de fabrication. De plus, dans ce type de test les circuits sont sur-testés ou sous-testés puisqu'il n'y a pas de mesure quantitative de qualité permettant de disposer d'un critère d'arrêt. Il en résulte qu'afin de minimiser le coût de test de production, il est nécessaire de développer des techniques de génération de tests basées sur des modèles de défaut. Dans les circuits analogiques les défauts sont classifiés comme paramétriques s'ils altèrent les performances du circuit, ou catastrophiques s'ils causent la perte totale du bon fonctionnement du circuit. Nous nous proposons donc, dans cette thèse, de mettre au point une méthode de test pour les circuits analogiques linéaires basée sur ces modèles de défaut. Cette méthode vise à générer un ensemble minimal de tests robustes permettant d'assurer à chacun des composants (paramètres) du circuit une détectabilité maximale de défauts. Cet ensemble de tests permettra de distinguer les circuits défectueux de ceux qui sont bons. Un circuit est déclaré défectueux si un test parmi l'ensemble de tests sélectionnés amène une performance du circuit en dehors de sa plage de tolérance. Un ensemble de tests est dit robuste s'il est en mesure de détecter les défauts paramétriques et catastrophiques sous la contrainte d'effet maximal de masquage dû à la tolérance des paramètres du circuit. En effet, seulement dans ce cas, la qualité d'un test peut être correctement mesurée et garantie. La génération d'un ensemble de tests ayant de telle spécification en particulier, et le test des circuits analogiques en général, sont des problèmes difficiles. Cette difficulté émane de la complexité naturelle de ces circuits qui présentent un spectre continu de défauts. La variation continue dans le temps des quantités physiques qui caractérisent ces circuits est à l'origine de la conti-

nuité de ce spectre. Par conséquent, il y a absence de modèle adéquat équivalent au modèle numérique "collé à 0 ou à 1", puisque les sorties analogiques ne peuvent être considérées comme des états hauts ou des états bas. En outre, le manque d'accessibilité aux noeuds internes et plus particulièrement la non-linéarité des caractéristiques de ces circuits, ainsi que l'effet de masquage dû au problème de tolérance, rendent le problème de test analogique plus complexe.

Tel qu'il a été mentionné ci-dessus, nous assistons aujourd'hui à une prolifération sans précédent des applications des circuits analogiques et mixtes. Il est donc urgent de pouvoir tester efficacement les produits de ces applications. Autrement, leur fiabilité est compromise.

1.1 Objectifs de la thèse

Dans cette thèse, nous nous proposons de mettre en oeuvre une méthode de génération de test multifréquences pour détecter les défauts paramétriques et catastrophiques dans les circuits analogiques linéaires. Etant donné un ensemble de performances d'un circuit et une plage de fréquences, l'objectif visé est de déterminer les valeurs des plus petites déviations (en valeur absolue) observables d'un paramètre défectueux du circuit, et de sélectionner les fréquences de test qui maximisent la détectabilité de ces déviations sous les contraintes de variations, dans le pire des cas[1], des autres paramètres dans leur plage de tolérance.

1. le pire des cas correspond à la configuration où les valeurs prises par les autres paramètres produisent un effet maximal de masquage empêchant les déviations du paramètre défectueux d'être détectées jusqu'à une certaine limite.

L'originalité de cette thèse réside principalement dans la génération de tests robustes de qualité garantie. En effet, cette garantie trouve son origine dans le fait que l'approche proposée permet de déterminer d'une manière optimale les valeurs des plus petites déviations (en valeur absolue) des paramètres du circuit à partir desquelles toute déviation de ces paramètres est assurée d'être détectée. En outre, elle permet de sélectionner la meilleure fréquence qui maximise l'observation (détection) de ces valeurs limites. L'originalité de cette approche émane aussi de la formulation du problème de génération de tests comme une série de problèmes d'optimisation. Enfin, cette originalité trouve sa racine dans la mise en oeuvre d'une méthode, fondée sur la programmation logique à contraintes, qui permet d'obtenir une solution globale à ces problèmes d'optimisation.

1.2 Analyse du problème

Dans l'univers analogique, une fonction élémentaire est réalisée par un groupe de composants interconnectés. A titre d'exemple, un transistor peut participer à la génération de plusieurs fonctions différentes dépendamment des composants avec lesquels il est interconnecté. A cause des fluctuations des paramètres du procédé de fabrication, un composant est réputé bon si sa valeur est comprise dans la plage de tolérance autour de sa valeur nominale. Il est important de mettre l'accent sur le fait que la valeur nominale n'est qu'une possibilité parmi une infinité de valeurs que peut

prendre un paramètre correct. Il en résulte que les effets de tolérance des paramètres induisent aussi un effet de tolérance sur la fonction (réponse de sortie, performance) associée. Par conséquent, deux circuits réputés bons réalisant la même fonction peuvent produire des réponses différentes à une même excitation, tout en restant dans la plage de tolérance. Autrement dit, les variations des paramètres se traduisent par des effets de variations des performances du circuit. Il s'en suit qu'un circuit est déclaré défectueux si la valeur de l'une de ses performances se trouve en dehors de sa plage de tolérance. Par conséquent, toute déviation d'un paramètre qui amène l'une des performances du circuit en dehors de sa plage de tolérance, est détectable en observant (mesurant) cette performance. Si l'on choisit un ensemble de performances tel que chaque paramètre du circuit est couvert par au moins l'une d'elles, alors l'observation de ces performances nous permet de distinguer les circuits défectueux de ceux qui sont bons. Naturellement, à ce stade plusieurs questions s'imposent à l'esprit parmi lesquelles: Selon quel critère faut-il sélectionner les performances à observer? Suivant quelle base peut-on sélectionner le test à appliquer? Comment garantir que les tests générés sont robustes (de meilleure qualité)?

Pour garantir la qualité d'un test, il faut nécessairement tenir compte des effets de tolérance des paramètres dans le pire des cas. En effet, supposons qu'un seul paramètre du circuit est défectueux (c'est une analyse de défaut simple). Et supposons aussi qu'on a pu observer l'effet de sa déviation en observant l'une des performances

du circuit sous la contrainte des variations des autres paramètres dans leur plage de tolérance dans le pire des cas. On en déduit que pour détecter cette déviation, il a fallu qu'elle soit suffisamment grande pour contrer d'une part, les effets de variations normales des autres paramètres dans le pire des cas et amener d'autre part, la performance observée en dehors de sa plage de tolérance. Autrement dit, le pire des cas correspond au masquage maximal, produit par les effets de tolérance des paramètres, qui empêche les variations d'un paramètre défectueux d'être détectées, à travers la performance observée, jusqu'à une certaine valeur limite au delà de laquelle toute déviation devient détectable. Il est clair qu'un ensemble de tests ne peut être robuste que s'il est en mesure de détecter les défauts paramétriques et catastrophiques sous la contrainte d'effet maximal de masquage dû à la tolérance des paramètres du circuit. C'est seulement dans ce cas que la qualité d'un test peut être mesurée et garantie. La génération d'un tel test s'avère un problème difficile puisque les circuits analogiques présentent un spectre continu de défauts. En outre, la non linéarité des performances de ces circuits complique davantage le problème du test analogique. La non linéarité se traduit par le fait que si la valeur d'un paramètre varie d'un certain facteur, les performances affectées par cette variation ne varient pas du même facteur. En d'autres termes, la relation $T_k = T(x_1,....,x_m)$ liant les performances T_k, k=1,...n, du circuit et les paramètres x_i,i=1....m, est non linéaire. Par conséquent, le problème du test analogique touche à un problème mathématique non linéaire multidimensionnel dans un espace continu, problème reconnu très complexe et très difficile mettant en oeuvre

beaucoup de temps de calcul. Dans cette thèse on se propose de formuler ce problème et de mettre au point une méthode d'optimisation globale pour le résoudre.

Avant de procéder à l'analyse et à la résolution du problème posé, il est important de passer en revue différents travaux accomplis dans le domaine du test analogique, afin de mieux situer le problème étudié dans cette thèse et de clarifier le cadre dans lequel il s'inscrit.

Chapitre 2
Revue de littérature

2.1 Concepts de base

La plupart des concepts de base utilisés dans le domaine du test des circuits analogiques sont empruntés à l'univers numérique. Dans cette section, nous rappelons certains concepts fondamentaux tout en mettant l'accent, quand c'est nécessaire, sur les spécificités qui caractérisent et distinguent l'univers analogique.

2.1.1 Test et diagnostic

Le processus de test peut être défini comme une procédure efficace d'extraction d'informations pertinentes concernant le système sous-test. En général, cette procédure est traduite par l'expérience dans laquelle le système est excité et sa réponse est analysée pour s'assurer de son bon fonctionnement. Si un comportement incorrect est détecté, alors l'objectif suivant est d'en localiser ou de diagnostiquer la cause et d'en déterminer la valeur.

2.1.2 Défauts physiques

Les défauts physiques sont à l'origine du fonctionnement incorrect d'un système. Ils sont détectés en observant les erreurs qu'ils produisent. Les défauts physiques peuvent être dûs à des erreurs de conception, des erreurs de fabrication, des défectuosités de fabrication ou des pannes physiques [5].

Les erreurs de conception sont dues à des spécifications incorrectes ou incomplètes, des violations des règles de conception, ou une incohérence entre les différents niveaux de conception. Les erreurs de fabrication peuvent être dues à une mauvaise sélection de composants, des connexions incorrectes ou à des court-circuits causés par des soudures mal faites. Par contre, les défectuosités physiques sont inhérentes à l'imperfection du procédé de fabrication, et ne sont pas par conséquent attribuables à des erreurs humaines. A titre indicatif, les court-circuits et les circuit-ouverts sont des défectuosités courantes du procédé de fabrication des circuits MOS LSI (Large Scale Integrated). Un profil de dopage incorrect et des erreurs d'alignement des masques sont aussi des exemples de défectuosités physiques. Il est à remarquer que la localisation précise des défectuosités physiques est un facteur pertinent permettant d'améliorer et de rendre mature un procédé de fabrication. Quant aux pannes physiques, elles peuvent se produire durant la vie d'un système et sont dues à l'effet de l'usure et à l'influence des facteurs d'environnement. À titre d'exemple, les connecteurs d'aluminium à l'intérieur d'un boîtier de circuit intégré s'amincissent au fil du temps et peu-

vent casser sous le phénomène de l'éléctromigration et de la corrosion. Les facteurs tels que la température, l'humidité et les vibrations contribuent à l'accélération du vieillissement des composants. Des radiations cosmiques et des particules α peuvent induire des pannes physiques dans les puces de mémoire à accès aléatoire (RAM) à haute densité. Certaines pannes physiques appelées "défauts de jeunesse" peuvent apparaître tôt après la fabrication.

Les défauts physiques peuvent être permanents s'ils sont toujours présents, ou intermittents s'ils apparaissent pendant certains intervalles de temps, ou transitoires (causés par exemple par un changement temporaire d'un facteur environnemental). En général, la considération des défauts physiques ne permet pas un traitement mathématique direct des problèmes de test et de diagnostic. La solution réside dans le traitement des quantités, maniables mathématiquement, et qui représentent les effets des défauts physiques sur les opérations du système. Ces quantités peuvent être des défauts logiques dans les circuits numériques. Alors que dans les circuits analogiques, elles peuvent être des défauts catastrophiques (similaire au modèle "collé à") ou des défauts paramétriques altérant les performances de ces circuits.

2.1.3 Modélisation

La modélisation joue un rôle primordial dans la conception, la fabrication et le test des systèmes électroniques. La façon avec laquelle nous représentons le système

conditionne la manière et les résultats de sa simulation pour vérifier sa conformité avec les spécifications. En outre, le modèle choisi pour représenter un système a également d'importantes conséquences sur la manière avec laquelle on construit les modèles de défauts, sur la simulation des défauts ainsi que sur la façon de générer les tests.

On distingue le modèle comportemental, le modèle fonctionnel et le modèle structurel. Le modèle fonctionnel d'un système numérique est une représentation de sa fonction logique (en l'occurrence sa table de vérité) abstraction faite des relations temporelles (diagrammes de temps). Le modèle comportemental est une combinaison du modèle fonctionnel et des relations temporelles. Cette distinction entre les deux modèles est importante du moment qu'elle permet à deux systèmes ayant la même fonction logique et des diagrammes de temps différents de partager un même modèle fonctionnel. Elle permet également aux deux aspects fonctionnel et temporel d'être traités séparément sur le plan vérification de la conception et génération de tests. Quant au modèle structurel, c'est une description hiérarchique dans laquelle chaque niveau est une collection d'éléments interconnectés, et tel que le niveau le plus bas est constitué d'éléments primitifs dont on connaît le modèle fonctionnel ou comportemental.

2.1.4 Simulation

Traditionnellement, les concepteurs ont utilisé un prototype pour la vérification d'un nouveau design. L'avantage principal d'un prototype est de pouvoir fonctionner à la vitesse de travail spécifiée. Par contre, il coûte cher et demande beaucoup de temps de mise en oeuvre. La simulation permet de remplacer le prototype par un modèle logiciel facilement modifiable et qu'on peut analyser sans difficulté. La simulation consiste à exciter le modèle avec une représentation des signaux d'entrée, et à déterminer l'évolution de la réponse de ce modèle en fonction du temps. Dans ce contexte, la simulation est une forme de test de vérification du design. Ce genre de test consiste à s'assurer que le design accomplit bien le comportement spécifié par comparaison des résultats obtenus par simulation à ceux attendus (spécifiés).

2.1.5 Modèle de défaut

Un modèle de défaut représente l'effet des défauts physiques sur le comportement d'un système modelisé. La modélisation des défauts physiques permet de réduire efficacement la complexité du problème d'analyse des défauts puisque plusieurs défauts physiques peuvent être modélisés par un même modèle de défaut. En outre, l'un des objectifs majeurs de la modélisation des défauts est de construire des modèles de défaut indépendants de la technologie utilisée. Autrement dit, un même modèle de défaut peut être appliqué à plusieurs technologies différentes. Il en résulte

que les méthodes de test et de diagnostic développées dans le cadre d'un modèle de défaut restent valables quand on change de technologie.

La modélisation de défauts est reliée étroitement au modèle construit pour représenter le système. En effet, on distingue les défauts[1] structurels dans le cas d'un modèle structurel, et les défauts fonctionnels dans le cas d'un modèle fonctionnel.

En général, dans l'univers numérique, les modèles de défauts structurels suppose que les éléments sont bons et que seulement les interconnexions peuvent être affectées. Les défauts typiques affectant les interconnexions sont des court-circuits et des circuit-ouverts. Dans l'univers numérique, le modèle structurel "collé à" est très répandu et a montré une très bonne efficacité représentative d'une large classe des défauts.

Par contre, dans le monde analogique il n'existe pas de modèle universel de défaut pour représenter les défauts analogiques. Différents facteurs sont à l'origine de ce sérieux handicap. En effet, les caractéristiques des circuits analogiques sont souvent non linéaires et le spectre de leurs valeurs est continu et assez large (à cause du problème de tolérance). Il en résulte que les méthodes déterministes sont souvent inefficaces pour modéliser ces circuits [6][7]. Les circuits analogiques sont aussi caractérisés par un spectre continu des défauts possibles. Par conséquent, il est très difficile de construire des modèles discrets simples et capables de couvrir tous (ou une large classe) les défauts possibles de ces circuits. En outre, les distributions statisti-

ques des défauts analogiques ne sont pas encore bien connues. Il s'en suit que les modèles probabilistes ne sont pas suffisamment efficaces.

Dans les circuits analogiques on distingue fondamentalement deux catégories de défauts: les défauts catastrophiques et les défauts paramétriques. Les défauts catastrophiques sont souvent le résultat des défectuosités aléatoires. Ces défauts peuvent être des déviations importantes des valeurs des composants ou des court-circuits et des circuit-ouverts [8]. Les défauts paramétriques sont causés par les fluctuations des paramètres du procédé de fabrication.

Aux difficultés mentionnés ci-dessus s'ajoute la question de savoir quel est le modèle de défaut qui domine et qu'on pourra utiliser? Malheureusement, il n'y a pas de consensus dans la communauté de test analogique concernant ce problème. En effet, certains travaux [9] mentionnent que 80 à 90% des défauts analogiques sont causés par des défauts catastrophiques tels que des composants (résistances, capacités, transistors) en court-circuits ou en circuit-ouverts, et que les défauts multiples sont peu probables (comme c'est le cas dans les circuits numériques). Par contre, d'autres travaux [10] mentionnent que la perte de rendement du procédé est causée par des phénomènes multiples. En particulier, la perte due aux défauts catastrophiques parait insignifiante dans le cas des circuits analogiques bipolaires. Quant au procédé BICMOS, il n'est pas encore suffisamment mature. Mais les premières études mon-

trent que les défauts multiples catastrophiques et paramétriques existent dépendamment de l'implementation de la circuiterie analogique [11].

2.1.6 Evaluation d'un test et simulation de défauts

Il est important de pouvoir évaluer l'efficacité et la qualité d'un test. L'évaluation d'un test est souvent effectuée dans le cadre d'un modèle de défaut. La qualité d'un test est mesurée par la couverture de défauts calculée comme le rapport du nombre des défauts que le modèle peut détecter au nombre total des défauts supposés possibles. L'évaluation d'un test se fait à l'aide de la simulation de défauts dans laquelle la réponse du modèle (représentant le circuit défectueux) est calculée. Un défaut est détecté s'il produit une réponse différente de celle produite par le modèle représentant le circuit réputé bon.

L'absence de modèle universel de défaut analogique, le problème de tolérance et la non linéarité des caractéristiques des circuits analogiques rendent les tâches de génération de tests, de diagnostic et de simulation de défauts extrêmement difficiles. Comme conséquence de la variation des paramètres du procédé, deux circuits, dont le fonctionnement est correct, peuvent produire des réponses différentes à une même excitation tout en restant dans la plage spécifiée. Cela provient du fait qu'à cause des fluctuations du procédé, les paramètres peuvent prendre une infinité des valeurs. Il en résulte, que contrairement au monde numérique, il n'est pas possible d'énumérer les

défauts analogiques possibles d'un modèle de défaut puisque leur spectre est continu. Par conséquent, compte tenu de la continuité du spectre des valeurs que peuvent prendre les paramètres bons (dans leur plage de tolérance) ou erronés (ayant dévié de cette plage), et de la non linéarité des caractéristiques (performances), un nombre extrêmement grand de simulations de défauts est requis pour assurer une évaluation correcte d'un test ou pour rendre efficace les méthodes de diagnostic basées sur la simulation de défauts (telle que l'approche de diagnostic à dictionnaire de défauts).

2.2 Méthodes de test

Compte tenu du fait que la plupart des principes des méthodes de test des circuits analogiques sont adoptés de l'univers numérique, il est intéressant de décrire brièvement les fondements de ces méthodes [5].

2.2.1 Méthode de test en temps réel (On-line testing)

Dans ce mode, le test est effectué dans les conditions normales de fonctionnement du système. Cette méthode suppose que le stimuli et la réponse du système ne sont pas connus à l'avance. En fait le stimuli est fourni en temps réel par les séquences générées par le système en fonctionnement. La méthode repose sur l'exploitation de certaines propriétés de la réponse, qui restent invariantes durant le fonctionnement correct (en l'absence de défauts) du système. Ces propriétés invariantes sont produites

par des techniques de design fiables permettant de les contrôler facilement durant le fonctionnement en temps réel. Les méthodes d'auto-vérification intégré (Built_In Self-Test) et d'auto-contrôle de design (*Self_checking design*) sont des méthodes de test qui peuvent être appliquées en temps réel.

2.2.2 Méthode d'auto-contrôle de design (*Self-checking design*)

Tel que mentionné précédemment, cette méthode est basée sur le contrôle de certaines propriétés invariantes de la réponse. Autrement dit, elle implique l'utilisation des codes de détection des erreurs (tel que le contrôle de parité, le code de Berger....) et des techniques spécifiques de design. La fonction de contrôle est assurée par un circuit de contrôle (*checker*).

2.2.3 Méthode d'auto-vérification intégré (*Built_In Self Testing*)

La méthode d'auto-vérification intégré (*Built_In Self_Test: BIST*) est une technique de design dans laquelle une partie du circuit est utilisée pour assurer l'auto-vérification du circuit. On distingue les différents types de BIST suivants:

-Le BIST concurrent en mode normal qui est une forme de test appliqué simultanément avec les opérations normales du système. Cette technique de test est souvent implementée en utilisant des techniques de codage ou bien la technique de duplication et de comparaison.

-Le BIST non-concurrent en mode normal est appliqué quand le système est dans un état de repos (état d'attente). Cette technique utilise des routines programmées (ou bien des routines ineffaçables écrites à l'usine) de diagnostic. Le processus de test peut être interrompu à tout moment pour permettre au système de rentrer en fonctionnement quand il est requis.

-Le BIST en mode test (*Off-line BIST*) met en oeuvre des générateurs de motifs de test (*Test-pattern generator: TPG*) et des analyseurs de réponse de sortie incorporés sur la puce ou la carte imprimée. Il va de soi que cette technique ne détecte pas les défauts en temps réel. Le BIST fonctionnel en mode test est basé sur une description fonctionnelle du circuit et utilise un modèle de défaut fonctionnel. Alors que le BIST structurel est basé sur l'usage d'un modèle de défaut structurel, et pour lequel la couverture de défauts est calculée sur la base des défauts structurels détectés.

Le BIST peut être utilisé pour tester les systèmes montés dans leur environnement de travail sans avoir recours à des équipements de test sophistiqués et coûteux. Cette méthode est basée sur la détection des défauts au niveau unités (telles que les cartes des circuits imprimés). A titre d'exemple, l'armée américaine a adoptée une politique de maintenance à deux niveaux [5]. Le système doit s'auto-tester pour détecter automatiquement l'unité défectueuse. Celle-ci est remplacée sur le terrain et l'unité défectueuse est rejetée ou envoyée pour réparation (second niveau de test).

2.2.4 Méthode de test à sonde guidée

C'est une technique utilisée au niveau des cartes imprimées. Durant la première phase du test, la détection des défauts se fait au niveau des broches de bord de la carte selon le test go/no go. Dans la deuxième phase, si le défaut est détecté le testeur décide quelle ligne interne de la carte doit être commandée, et demande à l'opérateur d'y placer la sonde. Ensuite, le test est appliqué de nouveau et la procédure continue de cette manière jusqu'à la localisation du défaut. Certains testeurs possèdent une monture sous forme de lit de clous permettant de commander toutes les lignes accessibles (en général les broches de circuit intégré) en une seule étape.

2.2.5 Méthode de test en circuit (*In-circuit testing*)

L'objectif de cette méthode est de tester les composants déjà montés sur un circuit imprimé. Son principe de fonctionnement est d'isoler le circuit intégré à tester du reste de son environnement et d'y appliquer des motifs d'entrée générés par un testeur externe. Ensuite, les réponses de ce circuit sont analysées.

2.2.6 Méthode de test par comparaison

Dans cette méthode la réponse du circuit sous-test est comparée à la réponse attendue (spécifiée). Celle-ci peut être générée par un circuit réputé bon (appelé unité

en or) ou bien par émulation en temps réel de l'unité sous-test. Cette méthode qui est basée sur la comparaison est en réalité utilisée par plusieurs autres méthodes de test.

2.2.7 Méthode de test par compression de la réponse

Cette méthode est basée sur la comparaison des signatures du circuit sous-test et celle de l'unité en or. En fait, au lieu de contrôler la réponse R du circuit sous-test, il s'agit de contrôler seulement une certaine fonction f(R) dérivée de cette réponse. La fonction f(R) est une représentation compressée appelée signature. A titre d'exemple, on peut compter le nombre des 1 (ou le nombre de transitions de 0 à 1 ou de 1 à 0) contenus dans la réponse du circuit sous-test et le comparer au nombre des 1 (ou des transitions) contenus dans la réponse du circuit réputé bon. Cette méthode est principalement utilisée dans la méthode d'auto-vérification intégré (BIST).

2.3 Modèle de défaut et simulation de défauts

Tel que mentionné précédemment, le manque de modèle analogique universel, le problème de tolérance et de non linéarité, de spectre continu de défauts rendent la simulation de défauts au niveau du circuit (c.à.d au niveau composants) très difficile, voir impraticable.

Idéalement, un circuit analogique peut avoir un ensemble infini de défauts. Afin de réduire cet énorme espace de défauts possibles et d'aboutir à une liste restreinte de

défauts qu'on peut simuler [12], il est nécessaire d'utiliser un simulateur de défectuosités tel que VLASIC (*VLSI Layout Simulation for Integrated Circuits*) [13] [14] qui est basé sur la technique d'analyse inductive de défauts (IFA) [15]. L'application de la technique IFA au dessin de masques (*Layout*) nécessite la connaissance des statistiques des défectuosités du procédé de fabrication. VLASIC utilise la méthode de Monte Carlo pour insérer les défectuosités du procédé dans le dessin de masques en respectant leur statistique. La technique IFA permet d'identifier une liste restreinte de défauts qui reflètent les effets du plus grand nombre possibles de défectuosités injectées (cent milles utilisées dans [16] et un million dans [17]). Bien que cette manière de procéder réduit considérablement le nombre de défauts à considérer, la liste de défauts obtenue peut être encore assez longue et le temps de simulation correspondant est important. Afin d'accélérer le processus de simulation et de diminuer en conséquence le temps de traitement requis, différents travaux ont été réalisés utilisant la modélisation à des niveaux d'abstraction plus élevés. Ces travaux sont classifiés en deux catégories. Dans la première, les défauts sont modélisés au niveau circuit, et le comportement de la macro-cellule défectueuse est saisi à un niveau d'abstraction approprié grâce à un simulateur analogique [12] [17] [18] [19] [20]. A titre d'exemple, dans [17] et [20] la macro-cellule (amplis opérationnels, comparateurs, filtres, etc...) contenant le défaut injecté, est décrite au niveau circuit (sous forme de liste de noeuds en SPICE), alors que les autres parties non défectueuses du circuit sont remplacées par leur modèle comportemental correspondant au niveau d'abstraction

approprié. Dans cette méthode, la plus grande partie du temps de simulation est consacrée à la partie sous évaluation (macro-cellule défectueuse). Il s'en suit que le temps de simulation devient presque indépendant de la taille du circuit.

Par contre, dans la deuxième catégorie, les défauts ne sont pas modélisés au niveau circuit. Le comportement d'une macro-cellule est saisi au niveau fonctionnel (Les valeurs des éléments du macro-modèle sont calculés à l'aide des équations de design qui peuvent être des équations différentielles et/ou des fonctions mathématiques décrivant les relations entre les composants du circuit) [21] [22]. De même, si les macro-cellules non défectueuses sont aussi simulées au niveau fonctionnel, le temps de simulation utilisant SPICE est réduit considérablement.

Les auteurs de [16] [23] ont développé une approche hiérarchique pour produire des modèles de défaut à des niveaux d'abstraction supérieurs en se basant sur l'amplificateur opérationnel comme élément primitif.

Pour modeliser les effets de variations des paramètres du procédé de fabrication certains auteurs ont utilisé une enveloppe, de valeur constante, autour des réponses nominales et des réponses des circuits défectueux [19] [24] [25]. Cela ne reflète pas nécessairement la réalité et peut affecter la précision des résultats puisque les valeurs limites constituant l'enveloppe, peuvent varier avec le test appliqué, le degré de non-linéarité des réponses du circuit, etc.

2.4 Design orienté testabilité (*Design For Testability: DFT*)

Le coût de test d'un système est devenu une composante majeure du coût de design, de production et de la maintenance d'un système. Il est le reflet de plusieurs facteurs tels que le coût de génération de test, le temps requis pour le test, le coût de l'équipement de test automatique (ATE), etc... En outre, pour minimiser ce coût il est important de détecter les défauts le plus tôt possible [26]. En effet, un même défaut détecté au niveau carte coûtera beaucoup plus cher que s'il décelé au niveau composant. Il coûtera encore plus cher s'il est détecté chez le client.

La technique de design orienté testabilité (DFT) a connu un essor important ces dernières années. L'objectif de cette technique est de réduire le coût de test en introduisant des critères de testabilité suffisamment tôt dans les différentes étapes du design. L'objectif fondamental de la technique DFT consiste à augmenter la contrôlabilité et l'observabilité du circuit de telle manière à en assurer la testabilité. La contrôlabilité est la facilité de contrôler un noeud du circuit et de lui attribuer une certaine valeur. Alors que l'observabilité est la facilité de propager un défaut d'un noeud vers une sortie primaire. La mesure de la contrôlabilité et de l'observabilité dépendent du générateur de vecteurs de test utilisé. Par exemple, dans le cas d'un générateur aléatoire un noeud a une faible contrôlabilité si l'établissement de son état nécessite une longue séquence de vecteurs d'entrée. Un circuit a une faible

observabilité aléatoire s'il nécessite une longue séquence de vecteurs d'entrée pour pouvoir propager l'état d'un ou de plusieurs noeuds vers les sorties du circuit.

Par contre, dans l'univers analogique, il n'est pas aussi simple de contrôler et de propager des signaux (et d'effectuer les simulations et les calculs associés) comme c'est le cas dans les circuits numériques. Cela provient de la complexité naturelle des circuits analogiques qui présentent des problèmes de tolérance, de non linéarité de leurs caractéristiques et de la continuité de leurs spectres de défauts.

Plusieurs travaux accomplis lient la mesure de testabilité des circuits analogiques au degré de solvabilité du système d'équations de diagnostic décrivant les relations entre les performances mesurées et les paramètres du circuit [27] [28] [29]. Dans [30] l'observabilité d'un défaut associé au composant x_i est définie comme la sensibilité d'une performance T_i du circuit aux variations de la valeur de ce composant. Elle peut être déterminée par le calcul de la sensibilité de la performance (fonction de transfert par exemple) au noeud de test, relative à la variation de la valeur du composant considéré.

La plupart des méthodes DFT requièrent des modifications du circuit et affectent certains facteurs tels que la surface de silicium, le nombre de broches d'entrées-sorties et la vitesse de fonctionnement du circuit. Les valeurs de ces facteurs ont tendance à augmenter suite à l'usage de la technique DFT. Or, augmenter la surface et la complexité des puces VLSI a pour effet d'augmenter la puissance consommée, de

diminuer le rendement du procédé de fabrication, et d'augmenter ainsi le coût de ces circuits. Par conséquent, un compromis doit être trouvé permettant d'atteindre un bon rendement et de rendre bénéfique l'ajout de la circuiterie de DFT.

On distingue principalement deux méthodes DFT: l'approche structurée basée sur les techniques de balayage (*Scan techniques*) et l'approche ad hoc.

2.4.1 Approche ad hoc

L'approche ad hoc essaie d'apporter une solution spécifique au problème de test d'un circuit donné. Afin d'augmenter la contrôlabilité et l'observabilité de certains noeuds non accessibles, on procède à la modification du design. Cette modification est réalisé par le (re)partitionnement du circuit permettant ainsi d'accéder directement ou indirectement à ces noeuds. La méthode de partitionnement la plus utilisée est basée sur le partionnement en blocs fonctionnels. On procède ensuite, à l'ajout des points de test (contacts métalliques) pour augmenter l'observabilité. Pour réduire le nombre des points de test et le nombre de broches associées on utilise le multiplexage. On peut aussi utiliser des commutateurs incorporés implementés avec des portes de transmission et des multiplexeurs permettant ainsi d'accéder à plusieurs noeuds au moyen des signaux de contrôle numériques [31] [32].

Par contre, il est plus difficile d'augmenter la contrôlabilité d'un noeud puisqu'il faut isoler l'entrée (noeud) de l'étage courant de la sortie de l'étage précé-

dent. Pour éviter d'effectuer l'isolation d'une manière destructive (couper la liaison au moyen des faisceaux électroniques par exemple) on procède à l'usage des portes de transmission et des multiplexeurs [31] [32]. L'ajout des commutateurs incorporés présente les inconvénients d'augmenter la surface et le temps de design. La nécessité des signaux de contrôle, ainsi que l'atténuation des signaux due aux portes de transmission et aux longs fils conduisant aux multiplexeurs et aux sorties primaires, constituent également d'autres désavantages.

Enfin, il est d'usage d'utiliser des interrupteurs analogiques permettant de briser les boucles des circuits analogiques pour pouvoir prendre convenablement les mesures. Cette façon de procéder permet d'éviter les coupures destructives des liaisons.

2.4.2 Approches structurées: Techniques de balayage (Scan techniques)

La méthode fondée sur la chaîne de balayage est plus répandue que la méthode ad hoc. Elle est appliquée durant la phase de design sous forme de directives à suivre ou comme des règles de design à respecter. Bien qu'il existe différentes méthodes, une technique de base pour augmenter la testabilité des noeuds consiste à les connecter à un registre de décalage. La question qui en découle est de savoir si l'on peut commander (à l'aide des signaux d'horloge) adéquatement le registre à décalage. Autrement dit, peut-on assurer la bande-passante, l'impédance, le niveau des signaux

nécessaires à la bonne conduite des mesures. Deux dispositifs bien connus qui fonctionnent comme un système d'échantillonnage et qui peuvent jouer le rôle des registres analogiques à décalage sont: le dispositif à transfert de charge (basé sur l'accumulation des charges dans des puits de potentiel) et le dispositif à transfert de charge réalisé à l'aide des transistors et des capacités (*Bucket_Brigade Device*). Ce dernier type de registre à décalage analogique a été utilisé dans [33] pour construire une chaîne de balayage analogique. Celle-ci est illustrée par la figure 2.1.

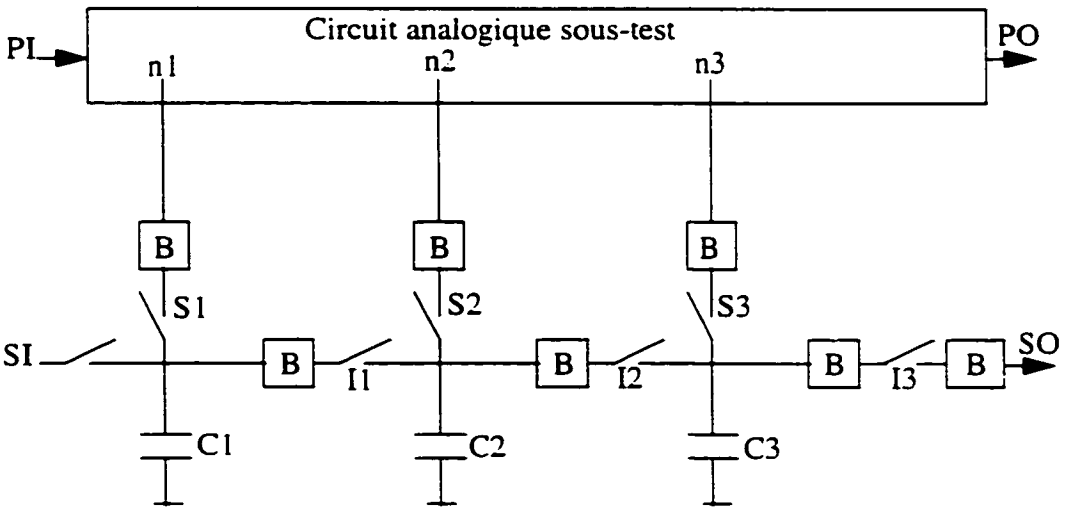


Figure 2.1: Chaîne de balayage analogique

Les signaux des noeuds n1, n2 et n3 sont échantillonnés grâce aux interrupteurs S1, S2 et S3, puis stockés dans les capacités C1, C2 et C3 qui sont isolées les unes des

autres par les suiveurs de tension B (amplificateur d'impédance d'entrée élevée et d'impédance de sortie faible). L'ensemble forme un échantillonneur-bloqueur. En mode normal, tous les interrupteurs sont ouverts coupant ainsi toute liaison entre les noeuds à tester et le registre à décalage. En mode test, les interrupteurs I1, I2 et I3 restent ouverts isolant ainsi les étages les uns des autres, alors que les interrupteurs S1 S2 et S3 se ferment pour permettre aux données des noeuds n1, n2 et n3 d'être échantillonnés en parallèle et stockées dans les capacités C1, C2 et C3. Ensuite l'interrupteur I3 est fermé (I2 et I1 restent ouverts) pour permettre au signal stocké dans C3 d'être transféré à la sortie SO. Puis, I2 est fermé (I1 reste ouvert) permettant au signal stocké dans C2 d'être transféré à son tour à la sortie. Enfin, le signal de C1 est transféré vers la sortie d'une façon similaire et l'opération entière se répète d'une manière cyclique. Durant ces opérations, la commande des interrupteurs peut être assurée par un simple registre à décalage numérique.

Cette chaîne de balayage analogique introduit des capacités parasites (intrinsèques aux amplificateurs) limitant ainsi la réponse en fréquence, plus particulièrement dans le cas des signaux à fréquences élevées. En outre, la circuiterie additionnelle peut être assez importante dépendamment du nombre des points de test requis.

Une autre classe de DFT utilise la technique d'auto-vérification (ou self-test intégré) (*Built-in Self-test: BIST*). En effet, les auteurs de [34]ont développé une technique BIST basée sur le test fonctionnel et la conversion des valeurs des performances

à observer en tensions continues permettant ainsi de rendre le test plus facile. Pour ce faire, la méthode utilise un générateur externe pour générer les tests (signal sinusoïdal de fréquence variable) requis. Ces tests sont sélectionnés de manière à maximiser la sensibilité des performances aux variations des composants du circuit. Les entrées et les sorties des blocs fonctionnels du circuit sont connectées à un multiplexeur analogique, qui à son tour est relié à un analyseur de réponse. Le stimuli est appliqué seulement au bloc fonctionnel sous-test. L'analyseur de réponse convertit la valeur de la performance mesurée (gain, phase,....) de la sortie du bloc sous-test en une tension continue. Cette tension est appliquée à un circuit de comparaison permettant de la comparer aux valeurs limites ($Vref_{min}$, $Vref_{max}$) de la plage de tolérance de la performance en question. Le résultat de la comparaison est stockée dans un registre à décalage numérique dont l'analyse permet de décider si la performance observée réussit ou non le test appliqué. Cette technique BIST présente certains inconvénients: le générateur de stimuli est externe et la circuiterie additionnelle peut être importante dépendamment du nombre des performances (impliquant autant d'analyseur de réponse) nécessaires pour couvrir l'ensemble des composants du circuit.

Les auteurs de [35] ont élaboré une technique BIST consistant à utiliser un générateur de stimuli en tensions continues (DC) intégré, un circuit de détection d'erreur et un multiplexeur analogique permettant de sélectionner le mode normal ou le mode test. Cette technique permet de détecter les défauts catastrophiques et

paramétriques qui affectent la fonction de transfert statique du circuit. Le circuit de détection d'erreur est conçu spécifiquement pour chaque circuit à tester. Ce circuit est construit autour d'un amplificateur opérationnel de manière à délivrer un signal d'erreur nul en absence de défauts ou un signal différent de zéro si le circuit est défectueux. Cette technique BIST permet d'observer plusieurs noeuds intermédiaires en même temps. En effet, le signal d'erreur fourni par le circuit de détection est une différence arithmétique entre un signal combinaison des signaux à observer (signaux de sorties des blocs fonctionnels) et un signal de contrôle. Celui-ci est calculé de telle façon que le signal d'erreur est nul en l'absence de défauts dans le circuit et non nul dans le cas contraire. En réalité, quand le circuit ne comporte pas de défauts, le signal d'erreur n'est pas exactement nul à cause du problème de tolérance. Pour tenir compte de ce problème, les auteurs ont introduit un seuil dont la valeur est calculée à partir des simulations de Monte-Carlo sous variations normales des composants du circuit, et au dessus duquel la valeur absolue du signal d'erreur indique la présence d'un défaut. Cette technique BIST ne requière qu'une faible quantité de circuiterie additionnelle et assure une bonne couverture de défauts considérés. Cependant elle ne peut pas couvrir les défauts telles que les variations anormales de la capacité d'un condensateur (celui-ci se comporte naturellement en circuit-ouvert vis à vis du courant continu). Il en résulte que cette méthode doit être complétée par une technique BIST utilisant un stimuli dynamique.

Une technique récente de DFT convertit les blocs fonctionnels d'un circuit en circuits oscillants [36]. La fréquence d'oscillation peut être exprimée en fonction des valeurs des composants du bloc sous-test ou bien en fonction de ses performances. La sensibilité de cette fréquence aux déviations des composants peut être maximisée en sélectionnant une fréquence de test adéquate et une structure d'oscillateur appropriée. En mode test, chacun des blocs du circuit est transformé en un oscillateur. Cette transformation consiste à rajouter un circuit de retour dans la structure du bloc sous-test et d'ajuster les valeurs de ses éléments pour obtenir des oscillations soutenues. Un multiplexeur analogique permet de sélectionner à chaque fois une des sorties des blocs du circuit. Ensuite, la fréquence d'oscillation correspondante est mesurée et analysée par un équipement externe. Cette technique de DFT présente principalement l'avantage de supprimer la nécessité du générateur de tests.

2.2 Diagnostic

On distingue deux méthodes fondamentales de diagnostic: la simulation avant test (*Simulation-Before-Test (SBT)*) et la simulation après test (*Simulation-After_test (SAT)*) [6]. Les techniques SBT sont basées sur une approche de dictionnaire de défauts. En effet, elles consistent à déterminer la réponse du circuit défectueux (dont le défaut injecté est tiré à partir d'une liste de défauts) à un test d'entrée donné. Ce processus qui est basé sur la simulation de défauts permet d'établir une base de don-

nées appelée dictionnaire de défauts. Le diagnostic consiste alors à apparier la réponse courante du circuit sous-test avec l'une des réponses (stockées sous forme de signature) du dictionnaire. Il est bien connu que les méthodes SBT sont plus appropriées pour détecter les défauts catastrophiques. Certains travaux tiennent compte du modèle "défaut simple". alors que d'autres considèrent le modèle "défauts multiples". Leur efficacité varie d'une méthode à l'autre dépendamment du nombre de simulations requises, du type de simulateur utilisé et de la précision du dictionnaire de défauts construit [37] [38] [39] [40]. L'application des techniques SBT peut détecter et localiser les défauts catastrophiques et sont souvent limitées aux circuits de taille petite ou moyenne.

Les techniques SAT sont plus appropriées pour la détection et l'identification des défauts paramétriques. Elles supposent que la topologie du circuit est connue. Les méthodes d'identification de paramètres et les techniques de vérification de défauts font parties des techniques SAT. Les techniques d'identification permettent de déterminer les valeurs des déviations des paramètres sous l'hypothèse que suffisamment de mesures (de réponses de sortie) indépendantes sont disponibles [41] [42] [43]. Ces techniques sont classifiées comme linéaires ou non linéaires dépendamment de la nature des équations de diagnostic qu'ils résolvent. Les techniques de vérification de défauts supposent que seulement quelques paramètres peuvent être défectueux qu'elles se proposent d'identifier (c.à.d. de déterminer leur déviation) et que tous les

autres éléments du circuit varient dans leur plage de tolérance [44] [45]. L'auteur de [27] a été le premier à définir le concept de solvabilité du système d'équations de diagnostic. La solvabilité est égale à zéro si tous les paramètres peuvent être déterminés indépendamment les uns des autres. Il en résulte que le système d'équations admet une solution unique. Par contre, si la solvabilité est égale à 1, alors un paramètre doit être connu pour pouvoir déterminer les valeurs des autres paramètres. Quand la valeur de solvabilité augmente, la solvabilité du système d'équations de diagnostic diminue. En se basant sur l'analyse de sensibilité, les auteurs de [46] ont dérivé les conditions nécessaires et suffisantes pour diagnostiquer les défauts catastrophiques. D'après ces auteurs une approximation d'ordre plus élevée que celle du premier ordre est nécessaire pour réduire les erreurs de diagnostic des défauts paramétriques.

Les méthodes SAT comportent plusieurs variétés de techniques qui peuvent localiser un défaut simple ou multiple, catastrophique ou paramétrique. Cependant, leur complexité et le temps de calcul requis limitent souvent leur application.

2.3 Génération de tests

La génération de tests est le processus permettant de déterminer le stimuli nécessaire pour tester un système. La génération de tests destinée au test de vérification de design et le développement des programmes de test et de diagnostic nécessitent beaucoup d'efforts de mise en oeuvre. Si la génération de tests est basée sur un

modèle de défaut, les tests générés détectent (et possiblement localisent) des défauts spécifiques. Par contre, si elle est basée sur un modèle fonctionnel, les tests générés sont tels que si le circuit les réussit, cela montre que le circuit satisfait la fonction qui lui est spécifiée.

Il est important de faire la distinction entre la génération de tests pour la vérification de design qui a pour objectif de trouver les erreurs de conception d'un design, et la génération de tests pour la détection des défauts physiques dans un système fabriqué.

Les circuits analogiques sont traditionnellement testés par la vérification de leur fonctionalité. Ce type de test classique est largement répandue dans l'industrie jusqu'à nos jours. Il est généré empiriquement sur la base des spécifications du circuit. Il ne repose pas sur un outil informatique ou sur un algorithme bien défini, et par conséquent ne fait pas partie des outils automatiques de génération de tests (*Automatic Test Pattern Generation: ATPG*). Les tests sont effectués dans le domaine du temps ou dans le domaine de fréquences dépendamment de la performance mesurée. Un circuit réussit un test si la valeur de la performance se trouve à l'intérieur de la plage de tolérance spécifiée.

Le test fonctionnel classique ne repose pas sur un modèle de défaut impliquant les composants internes du circuit, et par conséquent il n' y a pas de mesure de qualité définie. Le coût de ce genre de test est important et peut atteindre 30% du coût total de

fabrication [47]. Bien plus, dans certaines applications 95% des coûts totaux de test sont dépensés dans le test des circuits analogiques et 5% seulement dans le test des circuits digitaux, alors que 95% de la circuiterie de la puce est numérique et 5% seulement est analogique[48]. En outre, des travaux récents ont montré que certains circuits comportant des défauts catastrophiques peuvent réussir les tests fonctionnels de production [49].

Rappelons qu'en général, on distingue dans les circuits analogiques deux catégories de défauts. Les défauts catastrophiques sont causés par d'énormes variations des valeurs des paramètres tels que les composants en court-circuits ou en circuit-ouverts. Ils causent souvent une perte totale du bon fonctionnement du circuit. Les défauts paramétriques sont causés par des variations anormales des valeurs des paramètres et provoquent l'altération des performances des circuits.

Un algorithme de génération de test pour la détection des défauts catastrophiques sous les variations normales des paramètres a été mis en oeuvre par Milor et al. [50]. Etant donné une liste de défauts et un ensemble de tests fonctionnels, l'algorithme sélectionne un sous-ensemble de tests optimal qui assure une couverture de défauts spécifiée. Dans une première étape, l'algorithme sélectionne la performance (test) qui dévie le plus de sa bonne valeur sous l'influence de chacun des défauts de la liste courante des défauts. La performance choisie est ajoutée à la liste des tests sélectionnés T. Les valeurs limites de chacun de ces tests sont calculées à l'aide de l'ana-

lyse des variations en tenant compte de la tolérance des paramètres. L'ensemble de toutes les valeurs limites de l'ensemble T, constitue une région appelée par les auteurs la signature du bon circuit. D'une manière similaire, on construit la signature du circuit défectueux associé à l'un des défauts de la liste courante. Si la signature du circuit défectueux est différente de la signature (région) du bon circuit, alors le défaut est déclaré détecté par T. Il s'en suit que le défaut est supprimé de la liste des défauts. Ensuite, l'algorithme boucle sur la première étape pour sélectionner une autre performance (test) qu'on ajoute à T. Puis, on calcule la signature qui correspond au nouveau ensemble T de tests. Ensuite, on calcule de nouveau les signatures des circuits défectueux (associés chacun à l'un des défauts) et on supprime les défauts détectables par T de la liste courante des défauts. L'algorithme continue ainsi, jusqu'à obtention d'une couverture de défauts satisfaisante. Cet algorithme a été amélioré ultérieurement dans [51] et [52] et étendu aux défauts paramétriques. L'approche est basée sur un modèle de défaut dérivé à partir des fluctuations statistiques du procédé technologique de fabrication. A partir d'un ensemble de tests fonctionnels, l'algorithme sélectionne un sous-ensemble de tests qui détectent les défauts paramétriques. La sélection se fait selon le critère de minimisation du temps de test de production. Pour ce faire, l'algorithme élimine les tests de faible couverture de défauts. Autrement dit, un test t_i est sélectionné si la probabilité qu'un circuit défectueux réussisse l'ensemble de tests n'incluant pas t_i, est maximale. Soit T l'ensemble des tests t_i sélectionnés assurant la couverture de défauts spécifiée. La seconde étape consiste à trouver l'ordre suivant

lequel les tests doivent être exécutés afin de minimiser le temps de test. Pour trouver cet ordre, l'algorithme effectue plusieurs permutations des tests de l'ensemble T, calcule le temps requis par chacune d'elles et sélectionne celle qui requière un temps minimal pour son exécution. Dans [53] le problème de génération de tests a été formulé comme un problème de programmation quadratique. Cette approche a été développée pour la détection des défauts paramétriques. Elle détermine les entrées d'excitation qui maximisent la différence quadratique entre la réponse d'un circuit réputé bon et celle du circuit défectueux. Dans cette approche, les paramètres sont pris à leur valeur nominale et par conséquent, ne tient pas compte du problème de tolérance. Les auteurs de [54] ont développé une technique de génération de tests pour la détection des défauts catastrophiques et paramétriques, basée sur l'analyse de sensibilité et l'analyse de tolérance dans le pire des cas. Cependant, l'analyse en fréquence n'a pas été considérée et le modèle utilisé est une approximation du premier ordre. Dans [55] les auteurs ont développé un outil automatique permettant le calcul de sensibilité à l'aide de la technique des réseaux adjoints (*adjoint network*) pour la conception des circuits résistants aux variations des paramètres et permettant la génération de tests. Cet outil est basé sur les travaux [54] et [34]. Il est destiné pour la détection des défauts catastrophiques et paramétriques sous l'effet de variations des paramètres dans leur plage de tolérance. La méthode présentée en [56] est basée sur l'usage de modèle de défaut comportemental et sur l'analyse de sensibilité. Pour trouver la meilleure fréquence de test, l'approche consiste à faire varier (perturber) la sensibilité

en fonction de la fréquence pour déterminer celle qui maximise l'erreur entre les réponses du circuit défectueux et celle du bon circuit. Les auteurs de [34] ont élaboré une technique de génération de tests à fréquences multiples (chaque excitation est une sinusoïde ayant une fréquence de test distincte) basée sur l'analyse de testabilité et le concept d'observabilité (celle-ci est définie en fonction de la sensibilité). Les fréquences de test sélectionnées sont celles pour lesquelles la sensibilité des performances en fonction des déviations des composants défectueux, est maximale. Il est à noter que les approches [56] et [34] ne tiennent pas compte des effets de tolérance des composants et que les fréquences de test sélectionnées peuvent être non optimales. Les auteurs de [57] ont mis en oeuvre une technique de génération de tests pour la détection des défauts catastrophiques. Le problème de test a été formulé comme un problème d'optimisation min-max incluant les effets de variation des paramètres dans leur plage de tolérance. La méthode est basée sur la linéarisation de la réponse de macromodèle de défaut et l'usage des techniques de programmation linéaire pour générer des tests en tensions continus (DC). Partant d'un ensemble fini de valeurs DC comme espace de recherche, l'algorithme en sélectionne un sous-ensemble de tests qui maximisent l'erreur entre les réponses du circuit réputé bon et le circuit défectueux. Le problème est converti en une série des problèmes de programmation linéaire qui sont résolus itérativement.

La méthode de génération de test développée dans [58] détecte les défauts paramétriques qui affectent la réponse des circuits analogiques dans le domaine du temps. Les auteurs proposent deux alternatives pour le signal d'entrée: le signal sinusoïdal ou le signal en rampe saturée. Dans le cas où le signal est sinusoïdal, l'approche consiste à déterminer pour chaque défaut la fréquence qui maximise l'erreur E entre la réponse en amplitude (ou en phase) du circuit défectueux et celle du bon circuit. Afin d'aboutir à un ensemble de tests minimal, seulement les fréquences pour lesquelles l'erreur E est supérieure ou égale à un certain seuil, sont sélectionnées. Dans la seconde alternative, le signal d'entrée est un échelon ayant un temps de montée fini. Il est modélisé par une rampe saturée. Il s'agit dans ce cas, d'examiner les paramètres de la réponse du circuit, en l'occurrence les valeurs du temps de montée, du délai et de l'amplitude en régime permanent, ainsi que la valeur maximale du dépassement "overshoot" (partie transitoire du signal avant régime permenant). L'analyse des réponses des circuits défectueux montre que les déviations de ces paramètres de leurs bonnes valeurs sont suffisamment importantes. Il s'en suit qu'ils peuvent être utilisés pour la détection des défauts. Cet article montre d'une manière heuristique que le signal en rampe est efficace pour le test analogique. En effet, selon les auteurs, ce signal constitue une manière naturelle d'effectuer la compression des tests puisqu'il peut se substituer à un ensemble de signaux sinusoïdaux (tests). Partant d'un signal en rampe avec une pente donnée, l'heuristique consiste à ajuster cette pente d'une manière itérative jusqu'à obtention d'une (ou de plusieurs) rampe(s) qui

maximisent le nombre de défauts détectés. Grâce à cette méthode, seulement deux rampes (de pente différente) ont été générées permettant de détecter 100% des défauts considérés. Cette technique doit être complétée par une approche qui permet de détecter les défauts catastrophiques. En outre, elle ne tient pas compte des variations des paramètres dans leur plage de tolérance.

La génération de tests basée sur le contrôle du courant d'alimentation (I_{DDQ}) s'est avérée efficace pour détecter les défauts structurels dans les circuits numériques en technologie CMOS. Dans cette approche le courant parcourant les rails d'alimentation V_{DD} ou de masse V_{SS} est contrôlé durant l'application des vecteurs de test. Ces circuits tirent un courant très faible durant l'état de repos. Par contre, en présence de défauts dans le circuit, la valeur de ce courant augmente significativement (de plusieurs ordres de grandeur) offrant ainsi une technique pour les détecter. Cette technique offre l'avantage d'une détection directe des défauts sans avoir besoin de les propager vers une sortie du circuit. En outre, elle permet de détecter certains défauts non détectées par les approches basées sur le modèle "collé à". Cette approche a reçu beaucoup d'intérêt durant ces dernières années en vu de son application au test des circuits analogiques et mixtes [59] [60]. Les auteurs de [61] ont décrit différentes applications de l'approche I_{DDX} (X=d, q, t désigne respectivement la mesure du courant dynamique, statique ou transitoire) au test analogique. Ils ont aussi montré que l'influence des variations du procédé de fabrication affectant le courant de repos d'un

circuit non défectueux est évaluée à ±5% de la valeur de ce courant, et que dans la plupart des applications la déviation limite du courant permettant de distinguer les bons circuits de ceux qui sont défectueux, doit être au moins égale à ±20%. La technique proposée dans [62] consiste à faire varier la valeur de l'alimentation, et à mesurer le courant d'alimentation pour une variété de signaux d'entrée (impulsion, rampe, échelon). En faisant varier la valeur d'alimentation, l'effet d'un défaut est amplifié permettant ainsi de rendre le test plus efficace. Appliqué au test des convertisseurs, cette méthode a produit une couverture de défauts de 87%. Les auteurs de [63] ont montré que l'application d'un signal en rampe aux noeuds de l'alimentation génère des courants de polarisation riche en information, qui peuvent être utilisés pour la détection des défauts analogiques (puisque ces courants sont fonction de la topologie et des valeurs des composants du circuit). Chacun des défauts d'une liste pré-définie est simulé et la signature correspondante est générée à partir de la réponse en courant pour construire un dictionnaire de défauts.

Cependant, d'après [64], le test basé sur le contrôle du courant d'alimentation (I_{DDQ}) peut causer une certaine perte de rendement induisant une perte économique non acceptable dans certains secteurs de l'industrie qui fabriquent des produits "grand public" (tels que les jeux électroniques. Les produits de haute fiabilité destinés, par exemple, aux systèmes satellites n'en font pas partie). En effet, dans certains cas, il arrive qu'un circuit réussisse tous les tests fonctionnels et échoue seulement le test

I_{DDQ}. L'analyse de ces cas montrent l'existence des défauts qui n'affectent pas le fonctionnement du circuit. Ces circuits sont classifiés comme "des circuits défectueux qui fonctionnent". Certains d'entre eux présentent une fiabilité à risque. Le rejet de l'ensemble de ces circuits est à l'origine de pertes économiques importantes.

La technique de génération de tests pseudo-aléatoire empruntée au test numérique est appliquée au test des circuits analogiques linéaires [65]. Cette méthode suppose que le circuit à tester est incorporé entre deux convertisseurs: l'un est numérique/analogique (CNA) à l'entrée du circuit, l'autre est analogique/numérique (CAN) à sa sortie. Un registre numérique à décalage avec rétroaction linéaire (LFSR) est utilisé comme générateur des vecteurs de test pseudo-aléatoires. ceux-ci sont convertis par le CNA en signaux analogiques pour être appliqués au circuit sous-test dont la réponse est numérisée par le CAN. L'analyse de signature du signal obtenu permet de détecter la présence de défauts dans le circuit.

Chapitre 3

Algorithme de génération de tests robustes pour les circuits analogiques linéaires

RÉSUMÉ

Nous avons vu dans le chapitre 1 que les circuits analogiques sont traditionnellement testés en vérifiant leur fonctionnalité. Et que dans ce genre de test, en plus d'être coûteux, les circuits sont sur-testés ou sous-testés puisqu'il n'y a pas de mesure quantitative de qualité permettant de disposer d'un critère d'arrêt. D'où la nécessité de développer des techniques de génération de tests basées sur des modèles de défaut, permettant de minimiser le coût de test de production. Rappelons aussi que la difficulté du test des circuits analogiques provient de plusieurs facteurs parmi lesquels le manque d'un modèle analogique universel (similaire au modèle numérique "collé à"), la continuité et la non-linéarité des caractéristiques de ces circuits. Ajouté à cela, le problème de tolérance des paramètres qui induit de l'incertitude sur ces caractéristiques.

Dans ce chapitre, nous proposons une méthode de génération de tests multifréquences pour détecter les défauts paramétriques et catastrophiques dans les circuits analogiques linéaires. L'importance et l'originalité de cette approche résident dans le fait que le problème de génération de tests est formulé comme une série de

problèmes d'optimisation qui tiennent compte des effets de masquage dans le pire des cas. Ces effets de masquage sont dûs aux tolérances des paramètres. De ce fait, la qualité des tests générés est garantie. En outre, la méthode permet de sélectionner la meilleure fréquence qui maximise la détectabilité des défauts du circuit sous test.

Les problèmes d'optimisation formulés sont résolus grâce à une méthode de programmation non-linéaire, en l'occurrence la programmation quadratique séquentielle (SQP), disponible dans l'outil de mathématiques MATLAB. La méthode de génération de tests proposée est illustrée à travers un exemple simple, puis appliquée à un circuit d'essai pris souvent comme tel dans la littérature, à savoir le filtre biquadratique. La validation des résultats est réalisée à l'aide d'une méthode basée sur la simulation HSPICE combinée avec la méthode de Monte-Carlo. La technique "Fault dropping" est utilisée pour réduire le nombre de défauts. La couverture de défauts du filtre biquadratique est également évaluée et commentée.

Ce travail a été publié en partie dans "International Mixed Signal Testing Workshop"[1], et en sa totalité dans la revue "Journal of Electronic Testing (JETTA)" [2]. Il est décrit en détail dans ce qui suit.

Optimization based Multifrequency Test Generation for Analog Circuits

A. Abderrahman[1], E. Cerny[2], B. Kaminska[1]

[1]Département de Génie Électrique et d'Informatique
Ecole Polytechnique de Montréal
C.P 6079, Succ. Centre-Ville;Montreal,Qc,Canada,H3C 3A7
[2]Département d'Informatique et de Recherche Opérationnelle
Université de Montréal
C.P 6128, Succ. Centre-Ville; Montréal, Qc, Canada H3C 3J7

Abstract

A robust test set for analog circuits has to detect faults under maximal masking effects due to variations of circuit parameters in their tolerance box. In this paper we propose an optimization based multifrequency test generation method for detecting hard and parametric faults in linear analog circuits. Given a set of performances and a frequency range, our approach selects the test frequencies that maximize the observability of a parameter deviation on a circuit performance under the worst masking effects of normal variations of the other parameters. Experimental results are provided and validated by HSpice simulations to illustrate the proposed approach.

3.1 Introduction

Analog testing is a difficult and expensive task. The difficulty stems from the fact that, unlike in digital circuits, the physical quantities of analog circuits vary over time in a continuous range. This implies a continuum of possible defects. In consequence, there is a lack of adequate fault models, since the output values of analog circuits can not be considered as either high or low levels as in the digital world where a large class of defects can be modeled by stuck-at-0/1 faults. Analog circuits are traditionally tested by verifying their function, which is known to be costly. Indeed, the estimated cost of analog testing may represent 30% of the total manufacturing cost [1]. To minimize the test time and thus the cost of production testing of analog circuits, test generation techniques based on fault-models are required.

In general, faults in analog circuits can be classified into hard and parametric faults. Hard faults are caused by catastrophic variations in parameter values such shorts and opens, and usually induce a complete loss of correct functionality. Parametric faults are caused by an abnormal deviation of parameter values and result in altered performance. Both types of faults have to be detected by a test set. Milor et al. [2] reported on a test generation algorithm for detecting catastrophic faults under normal parameter variations. In [3,4] an approach based on a statistical process fluctuation model was derived to select a subset of circuit specifications that detect parametric faults and minimize the production testing time. Test generation is formulated in [5] as a quadratic programming problem. This approach was developed for parametric faults and it determines an input stimulus x(t) that maximizes the quadratic difference of responses from the good and the faulty circuits with other parameters at their nomi-

nal values. A test generation approach for hard and parametric faults based on sensitivity analysis and tolerance computation was exposed in [6]. In this approach the worst case performance was expressed in terms of sensitivity and parameter tolerance, however, frequency analysis was not considered, and the model was a **linearization** obtained from first order partial derivatives.

The method presented in [7] is founded on a fault-model and sensitivity. For a given fault-list, perturbation of sensitivity with respect to frequency is used to find the direction toward the best test frequency. In [8] the authors derived a multifrequency test generation technique based upon testability analysis and a fault observability concept. The test frequencies selected are those where the output performance sensitivity is maximum with respect to faulty component deviation. In the above approaches [7-8] the masking effects due to variations of the fault-free components in their tolerance box **are not considered** and **the test frequencies may not be optimal**. A DC test generation technique for catastrophic faults was developed in [9]. This technique is fault-based and test generation is formulated as an optimization problem including the effects of parameter variations.

Any robust test set has to detect parametric and hard faults under maximal masking effects due to normal variations of parameters. Indeed, only in this case the quality of a test set may be correctly measured and guaranteed. In this paper we propose a novel test generation approach for detecting hard and parametric faults in analog circuits. This method is based on multifrequency testing which is, in general, more suited for subtle parameter variations than DC testing. As in [9], the test generation is formulated as a series of optimization problems taking into account the maximal mas-

king effects due to normal parameter variations. In general, the resulting optimization problem is highly non-linear and is solved iteratively.

The paper is organized as follows:

In the next Section, the proposed approach is outlined. A precise problem formulation is elaborated in Section 3. The test generation algorithm is presented in Section 4. Experimental results are reported in Section 5. Conclusions and a description of our future work appear in Section 6.

3.2 The proposed approach

3.2.1 Objectives

The purpose of our work is to generate the smallest set of robust tests that maximize the observability of a parameter deviation on a circuit performance under the worst masking effects of normal variations of the other parameters. This means that for each parameter we determine a) Its smallest absolute possible deviation outside which its detectability can be guaranteed under any variation of the other parameters within their tolerances, and b) the frequency of the input signal (the test) which guarantees this detection.

A circuit is declared faulty if a test of the test set produces a performance outside its acceptance range. In this case, the parameter deviation associated with the fault is said to be observable. We consider single parametric fault here.

3.2.2 Problem analysis

The number of tests depends on the input space, the output space (test points) and the performance space. First we have to select these spaces. In this paper, the input space consists of an extended range of operating frequencies. Multifrequency testing is more appropriate for parametric faults than DC testing which is more suitable for hard faults. For instance, an AC test is needed to detect a variation in a capacitance value. The output space may be obtained by partitioning a circuit into functional blocks. Each block output can be considered a possible test point. A performance space has to be selected depending on the selected input space. We thus assume that a set of performances such as gain, Q-factor, phase, cut-off frequency, etc., a set of test points and a frequency range are given. The goal of minimizing testing time may be viewed as minimizing the number of performances, the number of test points where these performances should be measured, and the number of frequencies at which they should be observed. This goal can be reached if we are able to answer the following questions. (1) How can we select a performance to be measured? (2) How can we choose a test point where this measurement should be performed? (3) Finally, what frequency should be selected for the best observation? Assuming that the answers to the first two questions are given, we address here the third question.

Let $T_k(f, x_1, \ldots, x_m)$, k=1,.....n, be a set of n performances of the circuit under test, where $\boldsymbol{x} = [x_1 x_2 \ldots x_m]^T$ is the vector of parameters of the circuit. Let T represent one of these n performances and $T_{max}(f)$ and $T_{min}(f)$ the extreme values of T under the normal parameter variations that determine the acceptance range of T at frequency f. We emphasize that each component (i.e., parameter) of a circuit should be covered

by at least one performance. It is obvious that some performances are more sensitive to variations of a parameter than others. Those performances that are the most sensitive to parameter variations should be selected as test performances. More precisely, a performance may be selected as a test performance if it detects the smallest absolute minimum observable deviation of at least one parameter x_i (i=1.....m) of a circuit at some frequency f.

3.2.3 Problem formulation

Let T(f, **x**)=T(f, x_1,..... x_m), k=1.....n, be the performances to observe, functions of frequency f in $[f_{min}, f_{max}]$, $\boldsymbol{x} = [x_1 x_2 \ldots x_m]^T$ the vector of parameters of the circuit, and $\boldsymbol{x}_n = [x_{1n} x_{2n} \ldots x_{mn}]^T$ the nominal value of **x**. Let the normal tolerance of a parameter x_i be the interval $X_i = [x_{il}, x_{iu}]$, i=1.....m, and the total possible range of values x_i can take be from the interval $[\underline{x}_{il}, \bar{x}_{iu}]$ such that $\underline{x}_{il} \le x_{il}$ and $\bar{x}_{iu} \ge x_{iu}$. Let $\boldsymbol{x}_l = [x_{1l} x_{2l} \ldots x_{ml}]^T$ and $\boldsymbol{x}_u = [x_{1u} x_{2u} \ldots x_{mu}]^T$, hence under normal circumstances $\boldsymbol{x}_l \le \boldsymbol{x} \le \boldsymbol{x}_u$, and $\boldsymbol{x}_l \le \boldsymbol{x}_n \le \boldsymbol{x}_u$.

3.2.3.1 Valid range determination

Let T_{min}(f, **x**) and T_{max}(f, **x**) be the extreme values of T at frequency f under the variation of $\boldsymbol{x} \in [\boldsymbol{x}_l, \boldsymbol{x}_u]$. These extreme values can be obtained as the solution of the following optimization problems.

For a given frequency f.

$$T_{max}(f) = \max_{x} T(f, x)$$

subject to $x_l \le x \le x_u$

and

$$T_{min}(f) = \min_{x} T(f, x)$$

subject to $x_l \le x \le x_u$

The envelope delimited by $T_{max}(f)$ and $T_{min}(f)$ constitutes the acceptance range of T.

3.2.3.2 Minimum observable parameter deviation:

Without loss of generality, let x_m be the parameter whose changes we wish to observe at performance T. Let $x_{m\text{-}min}(f) \in [\underline{x}_{ml}, x_{ml}]$ (resp. $x_{m\text{-}max}(f) \in [x_{mu}, \bar{x}_{mu}]$) be the smallest (resp. largest) value of x_m such that T(f) is outside of the interval $[T_{min}(f), T_{max}(f)]$ for all x_m in $[\underline{x}_{ml}, x_{m\text{-}min}(f)]$ (resp., for all x_m in $[x_{m\text{-}max}(f), \bar{x}_{mu}]$.

Graphically, we can visualize the various intervals of values of x_m as follows:

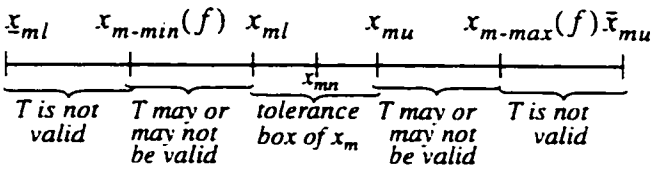


Figure 3.1: Total possible range of a parameter x_m under normal variations of x_i, $i \neq m$

Given some frequency f, the objective is to determine the values $x_{m\text{-}min}(f)$ (resp. $x_{m\text{-}max}(f)$) regardless the values of the other x_i, $i \neq m$, within their respective normal intervals. As a result, this gives us a limit on the deviation of x_m outside its normal tolerance box, such that if x_m is smaller than $x_{m\text{-}min}(f)$ (larger than $x_{m\text{-}max}(f)$), then this (faulty) deviation is guaranteed to be detected at f (i.e., T would be outside the range $[T_{min}(f), T_{max}(f)]$) independently of the other parameter values within their normal variations. Furthermore, we wish to find the frequency under which we can detect the largest value $x^*_{m\text{-}max}$ of $x_{m\text{-}min}(f)$ (resp. the smallest value $x^*_{m\text{-}min}$ of $x_{m\text{-}max}(f)$), i.e., the frequency which makes a faulty deviation of x_m most observable. This problem can be formulated as a max-min (resp. min-max) optimization problem as follows:

Let $\mathbf{x}_m$ be the parameter vector $\boldsymbol{x}_m = [x_1 x_2 \ldots x_{m-1}]^T$, and let δ be the resolution of the test equipment with respect to T.

Then

$$x^*_{m\text{-}max} = \underset{f}{MAX} \; \underset{\boldsymbol{x}_m}{MIN} \; (x_m)$$

subject to

$$T_{min}(f) - \delta \le T(f, \boldsymbol{x}) \le T_{max}(f) + \delta$$

$$x_{il} \le x_i \le x_{iu} \quad i = 1, \ldots, m-1$$

$$x_{ml} \le x_m \le x_{ml}$$

$$f_{min} \le f \le f_{ma}$$

and

$$x^*_{m\text{-}min} = \underset{f}{MIN} \; \underset{x_m}{MAX} \; (x_m)$$

subject to

$$T_{min}(f) - \delta \le T(f, \boldsymbol{x}) \le T_{max}(f) + \delta$$

$$x_{il} \le x_i \le x_{iu} \quad i = 1, \ldots, m-1$$

$$x_{mu} \le x_m \le \bar{x}_{mu}$$

$$f_{min} \le f \le f_{ma}$$

In general, these are complex non-linear optimization problems. We discretize the frequency in the interval of interest $[f_{min}, f_{max}]$ into a set $\{f_j\}$ of p frequencies and solve the MIN (resp. MAX) problem at each f_j, using the Sequential Quadratic Programming (SQP) method implemented in the "optimization toolbox" of Matlab [10]. The principal idea behind SQP is to transform the problem into a series of QP subproblems which are solved at each iteration. This method presents better performance (in terms of efficiency, accuracy and percentage of successful solutions) than every other method tested [11]. Although the SQP represents one of the best non-linear programming methods, T is needed to be one-dimensionally convex [10,11] in order to

guarantee that the global optimum is always found. However, if it is not the case, doing the optimization a number of times, each from different starting point may help to obtain the global optimum. When maximizing x_m, for example, the global maximum can be located if we start the optimization from points corresponding to a large value of x_m. Some other techniques (as scaling by variable transformation, using analytic partial derivatives instead of a finite difference approximation, etc.) are also useful for locating the optimum.

For each f_j we thus obtain the ranges $[T_{max}, T_{min}]$, $[x_{imin}^*, x_{imax}^*]$, i=1.....m, and we must select a subset of frequencies from$\{f_j\}$ that provide the best detection of a faulty deviation for each x_i.

3.3 Test generation algorithm

A simplified form of the algorithm is shown in below. The algorithm is given a set of performances $\{T_k\}$ and a set of frequencies $\{f_j\}$. In the first step, it finds the extreme values $T_{k_max}(f_j)$ and $T_{k_min}(f_j)$ of the performance T_k (k=1.....n) at each f_j. Then, each parameter value x_i (i=1.....m) is optimized at each frequency f_j. From the set of maximal (resp. minimal) values $x_{i_max}(f_j)$ (resp. $x_{i_min}(f_j)$), the minimum (resp., maximum) observable value $x^*_{i_min}(k)$,i.e., $x^*_{i_min}(k) = \min_{f_j} x_{i_max}(f_j)$ (resp., $x^*_{i_max}(k)$,i.e., $x^*_{i_max}(k) = \max_{f_j} x_{i_min}(f_j)$)of x_i, is extracted with the corresponding frequency $f_i^+(k) \in \{f_j\}$(resp. $f_i^-(k) \in \{f_j\}$) and the performance T_k. Finally, from the set of $x^*_{i_min}(k)$ (resp., $x^*_{i_max}(k)$), the smallest minimum (resp.,

the largest maximum) observable value x_i^S,i.e., $x_i^S = \underset{k}{min}\ \underset{f_j}{min}\ \underset{\mathbf{x_i}}{max}\ (x_i)$ (resp. x_i^G),i.e., $x_i^G = \underset{k}{max}\ \underset{f_j}{max}\ \underset{\mathbf{x_i}}{min}\ (x_i)$ is selected. At the same time, the corresponding performances T_k and the test frequencies of x_i^S and x_i^G are selected as (T_{ip}, tf_i^+) and (T_{iq}, tf_i^-), respectively. As a result, if d_i^+ and d_i^- are the smallest (resp. largest) relative observable positive (resp., negative) deviation,i.e., $d_i^+ = \frac{x_i^S - x_{in}}{x_{in}}$ (resp. $d_i^- = \frac{x_i^G - x_{in}}{x_{in}}$), each parameter x_i is characterized by two triplets (d_i^+, tf_i^+, T_{ip}) and (d_i^-, tf_i^-, T_{iq}), and for the whole circuit $D=\{(d_i^+, tf_i^+, T_{ip}), (d_i^-, tf_i^-, T_{iq}) | i=1,....,m\}$ defines a test set for all x_i. We can easily extract the set of test frequencies from D, while the additional information contained there identifies which x_i is tested at which T_k and frequency.

The obtained test set guarantees to detect any deviation of parameter x_i larger (resp. smaller) than the computed smallest minimum (resp. largest maximum) observable relative deviation d_i^+ (resp. d_i^-). On the other hand, the number of the observed performances and the test frequencies may be not minimal. There is always some trade-offs possible between the test quality and the testing time. This is, however, outside of the scope of this paper.

Simplified algorithm:

Begin

get $\{T_k \mid k=1,....,n\}$ and $\{f_j \mid j=1,....,p\}$

For k=1,....,n do

For all $f_j \in \{f_j\}$ do

$$T_{k_max}(f_j) = \max_{\boldsymbol{x}} T_k(f_j, \boldsymbol{x})$$

$$T_{k_min}(f_j) = \min_{\boldsymbol{x}} T_k(f_j, \boldsymbol{x})$$

End

For i=1,....,m do

For all $f_j \in \{f_j\}$ do

$$x_{i_max}(f_j) = \max_{\boldsymbol{x}_i} x_i$$

$$x_{i_min}(f_j) = \min_{\boldsymbol{x}_i} x_i$$

End

$$x^*_{i_min}(k) = \min_{f_j} x_{i_max}(f_j)$$

$f_i^+(k) = f_j$ /*/ f_j is such that $x^*_{i_min}(k) = x_{i_max}(f_j)$ *\

$$x^*_{i_max}(k) = \max_{f_j} x_{i_min}(f_j)$$

$f_i^-(k) = f_j$ /*/ f_j is such that $x^*_{i_max}(k) = x_{i_min}(f_j)$ *\

End;

End for k;

$D = \varnothing$

For i=1,....m do

$$x_i^S = \min_k x^*_{i_min}(k) \quad /*/ \; x_i^S = \min_k \min_{f_j} \max_{\mathbf{x_i}} (x_i) \; \backslash*\backslash$$

$T_{ip} = T_k$ /*/ k is such that $x_i^S = x^*_{i_min}(k)$

$tf_i^+ = f_i^+(k)$ /*/ k is such that $x_i^S = x^*_{i_min}(k)$

$$x_i^G = \max_k x^*_{i_max}(k) \quad /*/ \; x_i^G = \max_k \max_{f_j} \min_{\mathbf{x_i}} (x_i) \; \backslash*\backslash$$

$T_{iq} = T_k$ /*/ k is such that $x_i^G = x^*_{i_max}(k)$ *\

$tf_i^- = f_i^-(k)$ /*/ k is such that $x_i^G = x^*_{i_max}(k)$ *\

$$d_i^+ = \frac{(x_i^S - x_{in})100}{x_{in}}(\%)$$

$$d_i^- = \frac{(x_i^G - x_{in})100}{x_{in}}(\%)$$

$$D = D \cup \{(d_i^+, tf_i^+, T_{ip}), (d_i^-, tf_i^-, T_{iq})\}$$

End for m;

Extract test frequencies and build fault dictionary from D

End of algorithm.

3.4 Experimental Results

3.4.1 An illustrative example

To illustrate the above technique consider the low-pass filter shown in Figure 3.2. The performance of interest is the magnitude of the transfer function $|T(i\omega)|$:

$$|T(i\omega)| = \left|\frac{V_{out}}{V_{in}}\right| = \frac{R_2}{R_1}\frac{1}{\sqrt{1 + R_2^2 C^2 \omega^2}} \qquad (1)$$

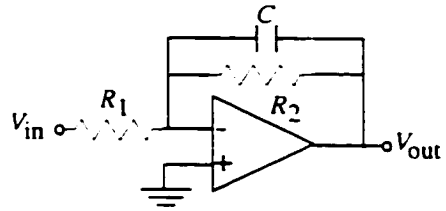


Figure 3.2: Low-pass filter

The nominal design (R_1=1.6k, R_2=16k, C=10nF) has the dc gain of 20dB, with the -3dB point at 1kHz. Figure 3.3 shows the computed results of the extreme values of $|T(i\omega)|$ under normal variations of R_1, R_2, and C in their tolerance intervals: R_1=[1.52, 1.68] $k\Omega$ R_2=[15.2, 16.8] $k\Omega$ and C=[9.5, 10.5] nF.

Figures 3.4, 3.5, and 3.6 show the computed maximum (minimum) values $x_{i_max}(f_j)$ (resp. $x_{i_min}(f_j)$) of R_1, R_2 and C observable at $|T(i\omega)|$, as functions of frequency. Table 3.1 shows the computed and simulated results. Indeed, the column "PV" (parameter vector) gives the minimum (resp. maximum) observable value $x^*_{i_min}$ (resp. $x^*_{i_max}$) of a faulty parameter (R_1, R_2, or C), and the values of the other parameters which produce the maximum masking of the faulty parameter observed at T. The smallest (resp. largest) observable positive (negative) parameter deviations are indicated in the column "S.O.P.D/L.O.N.D". To validate these results, an Hspice simulation was performed and each computed PV was simulated. Column " SFG" indicates the simulated value of the faulty gain produced by the parameter vector "PV" observed at the test frequency "TF". The column "Error" shows that the magnitude of the errors between the simulated faulty gain and the envelope (Tmin, Tmax) are very close to the tester resolution which is 0.01V. The set of tests for the low-pass filter is: TV={1Hz,40Hz,60Hz,6Khz,8Khz}. This set detects faulty circuit with R1 outside]1.37, 1.86[kOhm, R2 outside]13.74, 18.59[kOhm or C outside]8.5, 11.74[nF. From Table 3.1 we can see that any faulty deviation outside the range]-15, 17.4[% of any parameter is detected by the test TV. As a result TV achieves a fault coverage of 100% for deviation faults outside this range. In the range]-15,-14.1]% $\cup$ [16.2, 17.4[% only faults in R1 and R2 are detected by TV and the corresponding fault coverage is 66.6%. On the other hand, faults in the range]-14.1,-5[$\cup$]5,16.2[% are not guaranteed to be detected by TV.

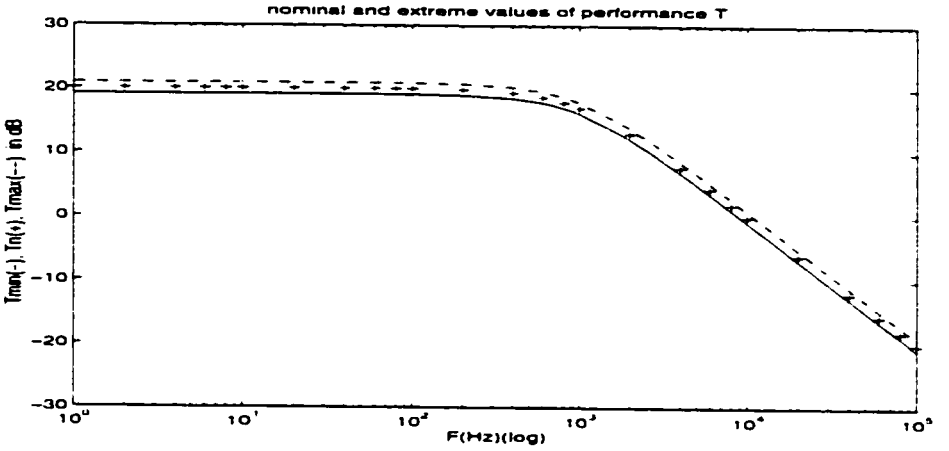


Figure 3.3: Extreme values T_{max}, T_{min} of filter gain

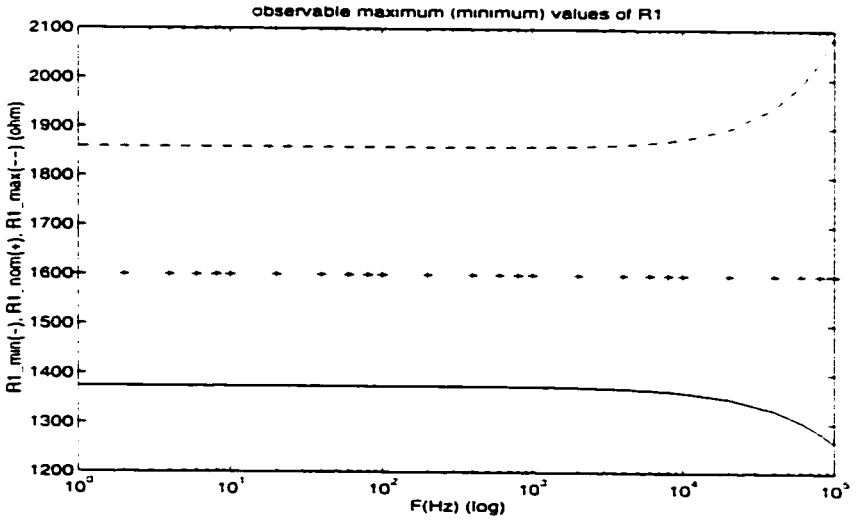


Figure 3.4: Observable minimum (maximum) deviation of parameter R_1

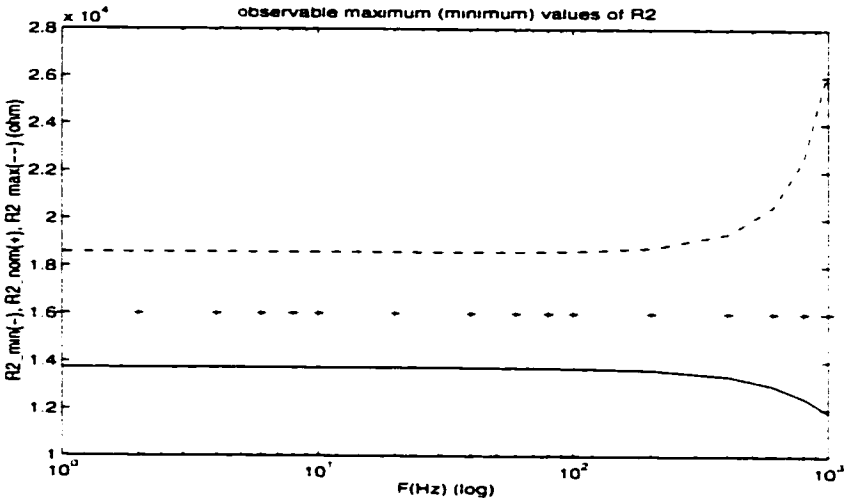


Figure 3.5: Observable minimum (maximum) deviation of parameter R2

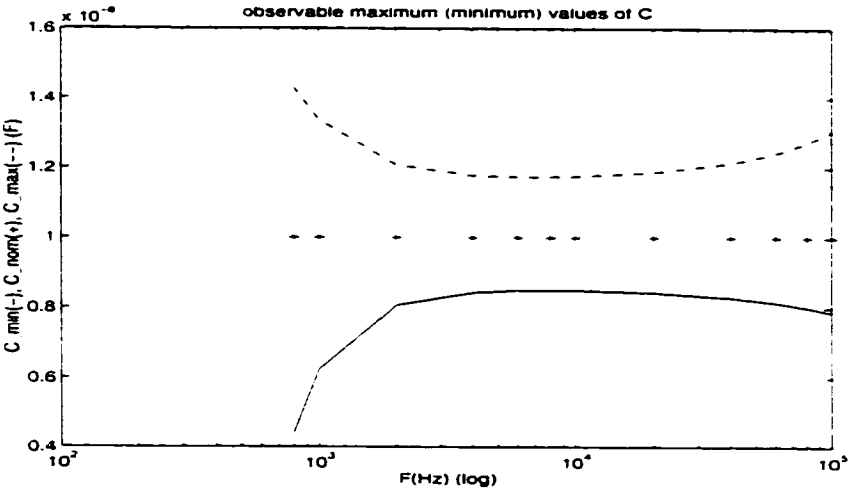


Figure 3.6: Observable minimum (maximum) deviation of parameter C

TABLE 3.1 Computed and simulated results

P	PV	S.O.P.D/ L.O.N.D (%)	TF (Hz)	T_{max} at TF	T_{min} at TF	S.F.G	Error
R_1	$R^*_{1_min}$ = 1.858k R_2=16.8k, C=9.5 nF	16.2	60	11.052	9.047	9.038	-0.009
	$R^*_{1_max}$ = 1.374k R2=15.2k, C=10.5nF	-14.1	40	11.052	9.047	11.063	0.011
R_2	$R^*_{2_min}$ = 18.585k R_1=1.68k, C=10.5 nF	16.2	1	11.052	9.047	11.063	0.011
	$R^*_{2_max}$ = 13.737k R_1=1.52k, C=9.5 nF	-14.1	1	11.052	9.047	9.038	-0.009
C	C^*_{min} = 11.739nF R_1=1.52k, R_2=16.8k	17.4	6k	1.812	1.483	1.473	-0.01
	C^*_{max} = 8.499nF R_1=1.68k, R_2=15.2k	-15	8k	1.367	1.119	1.378	0.011

3.4.2 A Realistic Application

As a realistic application for our test generation method consider the biquadratic filter shown in Figure 3.7. The performances (output responses) of interest are the magnitudes V_3 and V_5 of the transfer functions at nodes 3 and 5, given by the following equations:

$$V_3 = \frac{1}{R_6 C_2} \frac{\omega}{\sqrt{\left(\frac{R_8}{R_3 R_5 R_7 C_2 C_4} - \omega^2\right)^2 + \frac{\omega^2}{R_1^2 C_2^2}}} \quad (2)$$

$$V_5 = \frac{1}{R_3 R_6 C_2 C_4} \frac{1}{\sqrt{\left(\frac{R_8}{R_3 R_5 R_7 C_2 C_4} - \omega^2\right)^2 + \frac{\omega^2}{R_1^2 C_2^2}}} \quad (3)$$

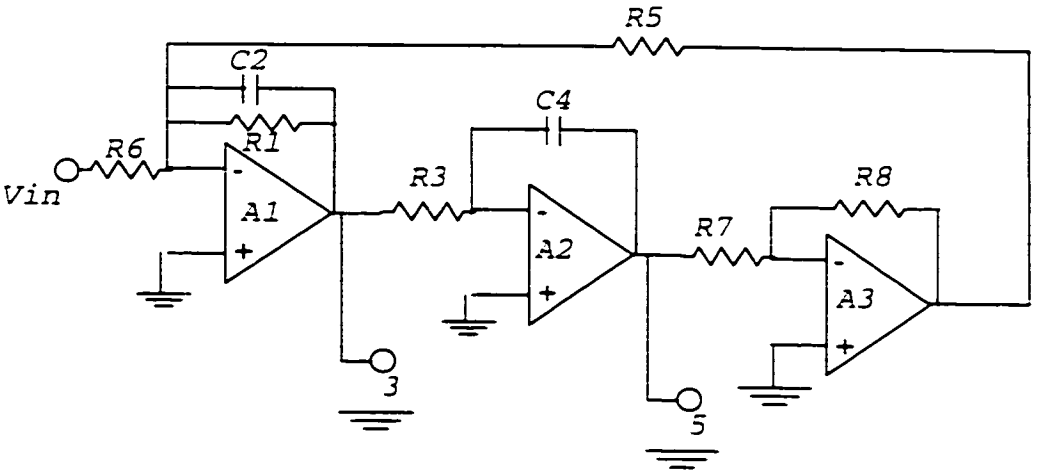


Figure 3.7: biquadratic filter

The frequency interval of interest is [1Hz.20Khz]. The smallest (largest) observable positive (negative) deviation of a parameter x_i depends on the transfer function. the tolerance intervals of the parameters. the resolution of the test equipment. the test frequency and other factors. In our case. the parameter tolerances and the tester resolution are assumed to be ± 5% and 0.01V. respectively. The results of the application of our method to the circuit are shown in Table 3.2. The third and the fourth columns give the smallest positive deviation "S.O.P.D" and the largest negative deviation "L.O.N.D"

observed at the outputs V5 and V3 respectively at the test frequency Fr (Hz). For each parameter the final selected test frequency TF and the corresponding test performance are indicated in the last column.

The results of Table 3.2 were validated in a similar manner as those in the previous example. Indeed, the parameter vector consisting of the minimum (resp., maximum) observable values $x^*_{i_min}(k)$ (resp., $x^*_{i_max}(k)$) of a faulty parameter x_i, and the values of the other parameters which produce the maximum masking of the fault is injected in the model of the circuit and then simulated. The error between the simulated faulty performance and the envelope (T_{k_max}, T_{k_min}) of the acceptance range is compared with the tester resolution. Simulations of all computed parameter vectors corresponding to the circuit parameters were performed, and the magnitude of the errors between the obtained performances and the normal envelopes were very close to the tester resolution. For illustration, some of the simulation results are shown in Figures 3.8 and 3.9. We can see that the envelope of the acceptance range is the region delimited by $V_{5_max}(f)$ and $V_{5_min}(f)$. The regions where the performance is observed at the selected test frequency are magnified to show the magnitude of the errors which are very close to 0.01V (the tester resolution value), confirming the correctness of the results obtained by our method.

TABLE 3.2 Computed results

para-meter	nominal value	Primary output V5		Primary output V3		Final test frequency TF [Hz]
		S.O.P.D/ L.O.N.D	Fr(Hz)	S.O.P.D/ L.O.N.D	Fr(Hz)	
R1	10k	43 -31.6	10k 9.6k	24.9 -23.3	10.300k 10.380k	10.3k (V3) 10.38k (V3)
C2	1.59nF	66.2 -59	20k 9.3k	34.93 -33.05	20 k 20k	20k (V3) 20k (V3)
R3	10k	45.7 -31.2	9.6k 11.52k	92.4 -45.2	4.020k 4k	9.6k (V5) 11.52k (V5)
C4	1.59nF	45.3 -30.3	9.6k 11.04k	92.1 -45	4.1k 4k	9.6k (V5) 11.04k (V5)
R5	10k	42.9 -30.5	1 1	92.1 -45.2	4k 4k	1 (V5) 1 (V5)
R6	10k	40.4 -29.1	8.140k 1	27.4 -17.42	11.680k 8.640k	11.680k (V3) 8.640k (V3)
R7	10k	42.9 -30.5	1 1	92.1 -45.2	4k 4k	1 (V5) 1 (V5)
R8	10k	43.5 -30.2	1 1	82.2 -48	4k 4k	1 (V5) 1 (V5)

The set of tests, extracted from Table 3.2, is:

$$S = \{8.64k, 10.3k, 10.38k, 11.68k, 20k\}_{V_3} \cup \{1Hz, 9.6k, 11.04k, 11.52k\}_{V_5}$$

S detects any fault deviation outside the following ranges (in%): R1 $\in$]-23.3, 24.9[, C2 $\in$]-33.05, 34.93[, R3 $\in$]-31.2, 45.7[, C4 $\in$]-30.3, 45.3[, R5 $\in$]-30.5, 42.9[, R6 $\in$]-17.42, 27.4[, R7 $\in$]-30.5, 42.9[, R8 $\in$]-30.2, 43.5[. The fault coverage thus reaches 100% for any fault deviation outside the range]-33.05, 45.7[% of any parameter. On other hand, in the range]27.4, 45.7[% only faulty deviations of R1 and R6 are guaranteed to be detected. As a result, the guaranteed fault coverage is only 25% which is poor. Consequently, for improving the fault coverage

the effect of the parameters R3, C4, R5, C2, R7, R8 should be observed at more sensitive performances. Furthermore, for selective filters, positive deviations of the parameter R1 become undetectable under other parameter variations. Even if we also observe the signal phase, only very large deviations of R1 can be detected. A general solution to such a problem of undetectability is the subject of on-going research. As mentioned earlier in section 2 and illustrated here by this example, an appropriate selection of performance and test point spaces combined with an adequate selection of test frequencies are the key to a complete solution for the test generation problem.

It is important to note that the size of the test set S may be greatly reduced by using "fault dropping". For example the faulty deviations of the parameter R3 (see Figure 3.10 which is described below) can be detected at frequency 20kHz which is already associated with the parameter C2. As a result, the frequencies 9.6 Khz and 11.52kHz associated with R3 can be eliminated. Similarly, one of the frequencies 10.3KHz or 10.380kHz can be dropped without any effect on the detectability of the faulty deviations of R1 (see Figure 3.12). Obviously, "fault dropping" may also reduce the fault coverage since the limits S.O.P.D and L.O.N.D may be altered with the reduced test set. So, by careful selection of the final test we can overcome this problem.

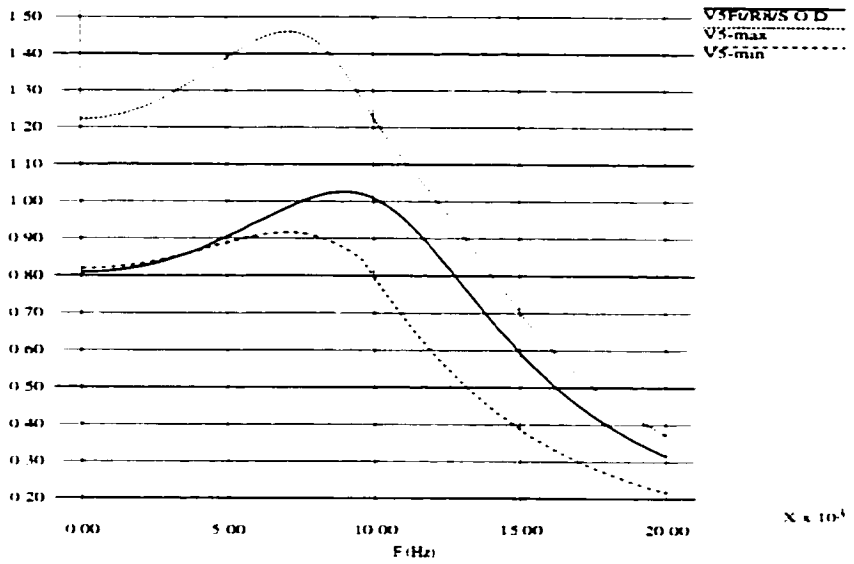
3.4.3 Test set validation

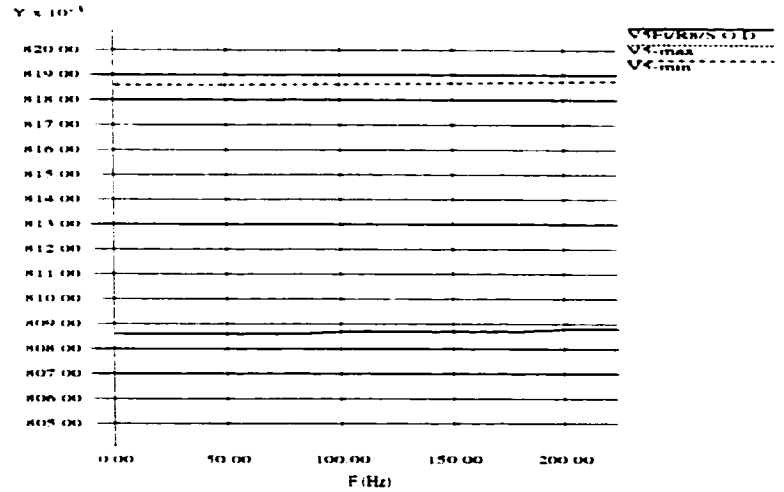
In order to validate the test set generated by our approach, we propose a fault simulation method based on faulty parameter deviations under the effects produced by fault-free parameter variations. Only single fault is considered at a time. The fault is injected by changing each parameter by -50%, the computed largest observable negative deviation, the computed smallest observable positive deviation, +50%, +100% and +1000%. Catastrophic faults (opens and shorts) are also considered.

The validation process consists of generating a family of curves of the good output responses (performances) at the observed point under random normal variations of the parameters, and another family of curves for the output responses under the faulty parameter and the other random normal variations of the fault-free parameters. The normal and the faulty families are compared to see if they are disjoint for at least one frequency from the test set for the injected fault.

The parameter variations are generated randomly. The distribution function associated with each parameter is assumed to be uniform and 100 iterations are performed in each HSpice simulation. Due to the large number of simulated curves, we show only some of them (samples of simulated faults for parameters R3, R8 and R1

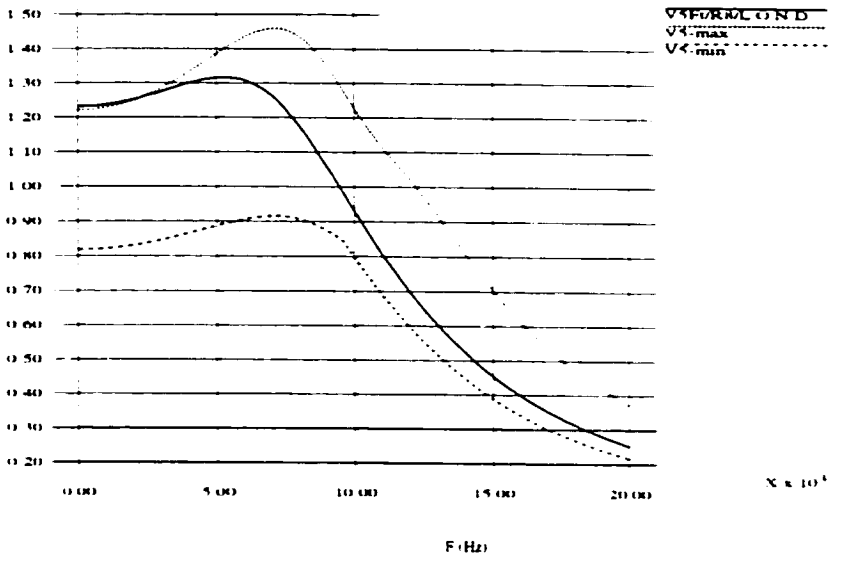
are given in Figures 3.10, 3.11 and 3.12, respectively). The simulation results show that all injected faults are detected by the test set, as predicted.

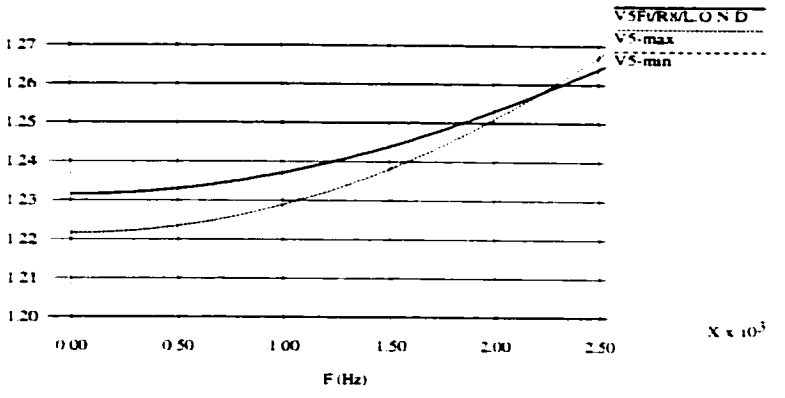




Magnified region where S.O.P.D of R_g is observed on V5Ft at the test frequency TF=1Hz.

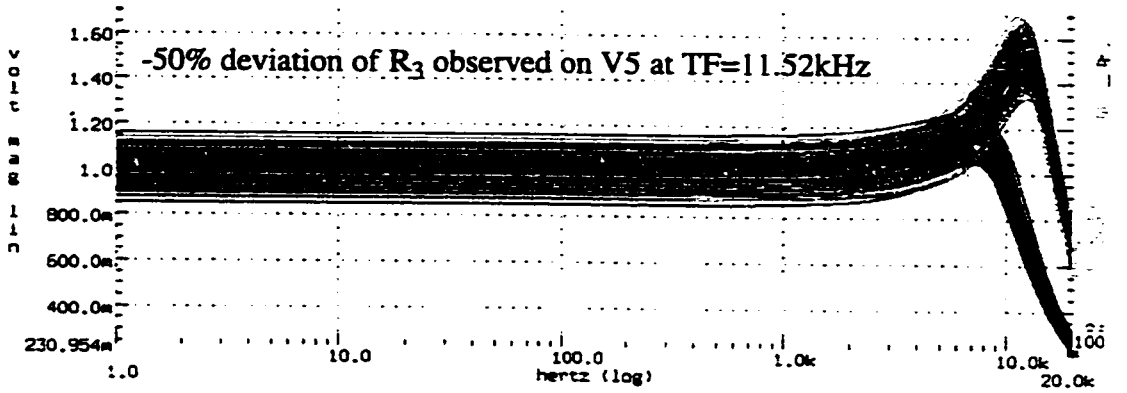
Figure 3.8: Faulty response V5Ft observed under the smallest observable positive deviation (S.O.P.D) of R_g, compared to the envelope of the normal range (V5-min, V5-max)

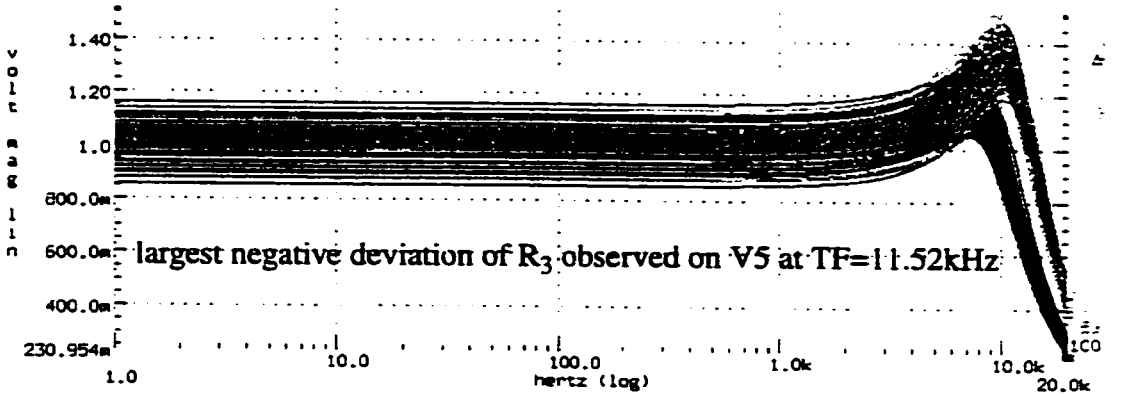


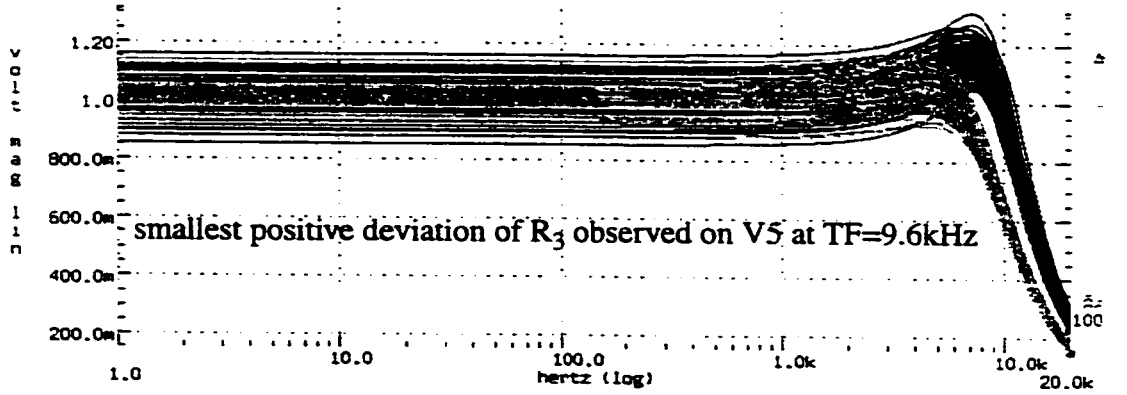


Magnified region where L.O.N.D of R_8 is observed on V5Ft at the test frequency TF=1Hz.

Figure 3.9: Faulty response V5Ft observed under the largest observable negativedeviation (L.O.N.D) of R_8, compared to the envelope of the normal range (V5-min, V5-max)


-50% deviation of R_3 observed on V5 at TF=11.52kHz
volt mag lin
1.60
1.40
1.20
1.0
800.0m
600.0m
400.0m
230.954m
1.0
10.0
100.0
1.0k
10.0k
20.0k
hertz (log)
100


largest negative deviation of R_3 observed on V5 at TF=11.52kHz
volt mag lin
1.40
1.20
1.0
800.0m
600.0m
400.0m
230.954m
1.0
10.0
100.0
1.0k
10.0k
20.0k
hertz (log)
100


smallest positive deviation of R_3 observed on V5 at TF=9.6kHz
volt mag lin
1.20
1.0
800.0m
600.0m
400.0m
200.0m
1.0
10.0
100.0
1.0k
10.0k
20.0k
hertz (log)
100

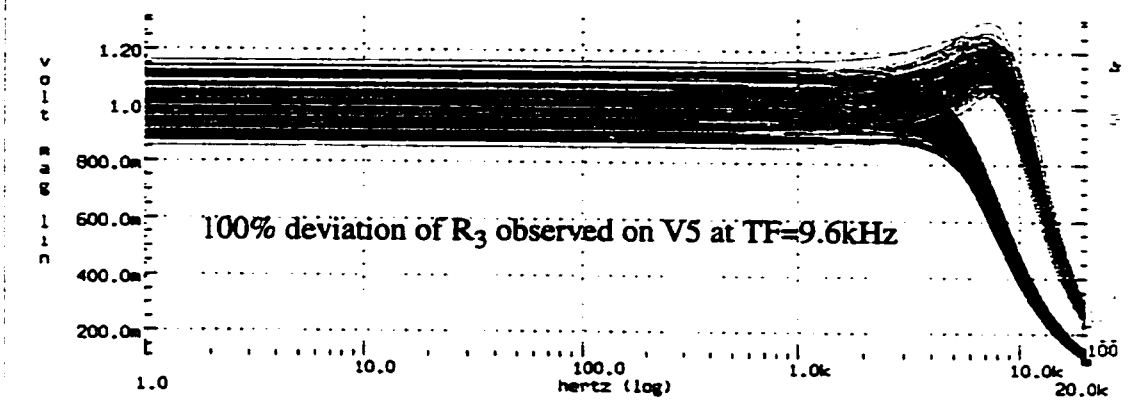
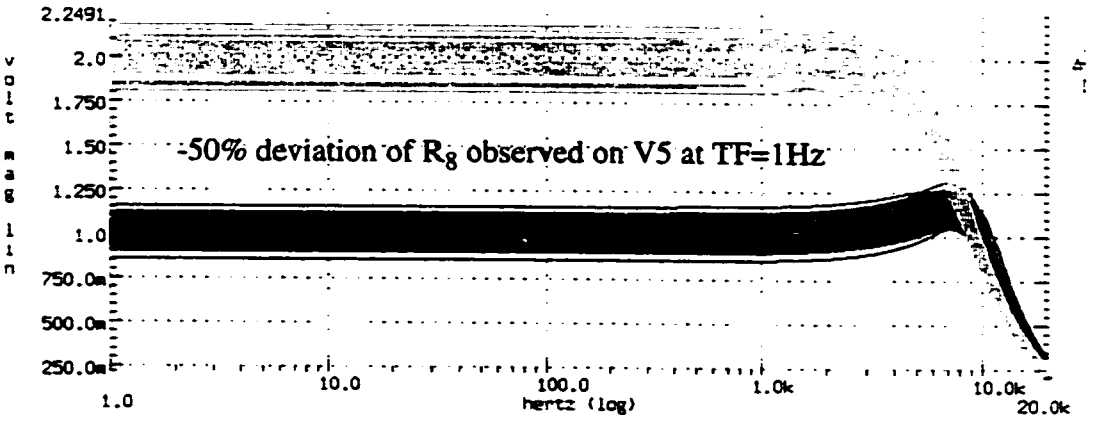
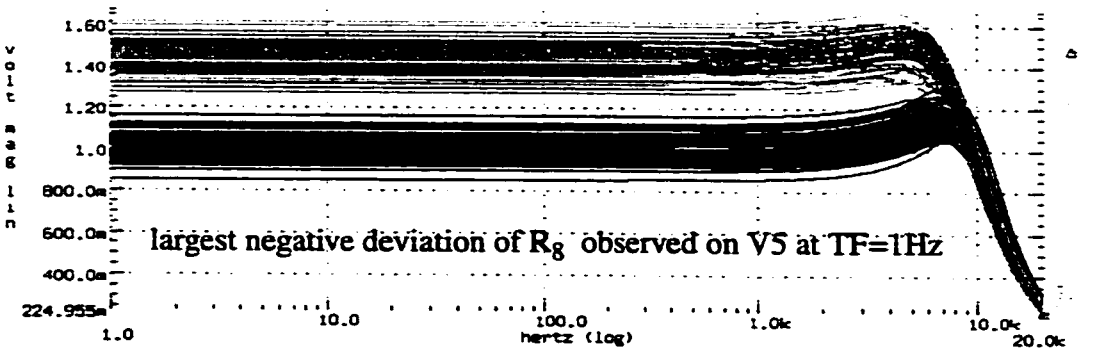


Figure 3.10: Faulty and good response families under faulty R_3 and the other parameters varied randomly





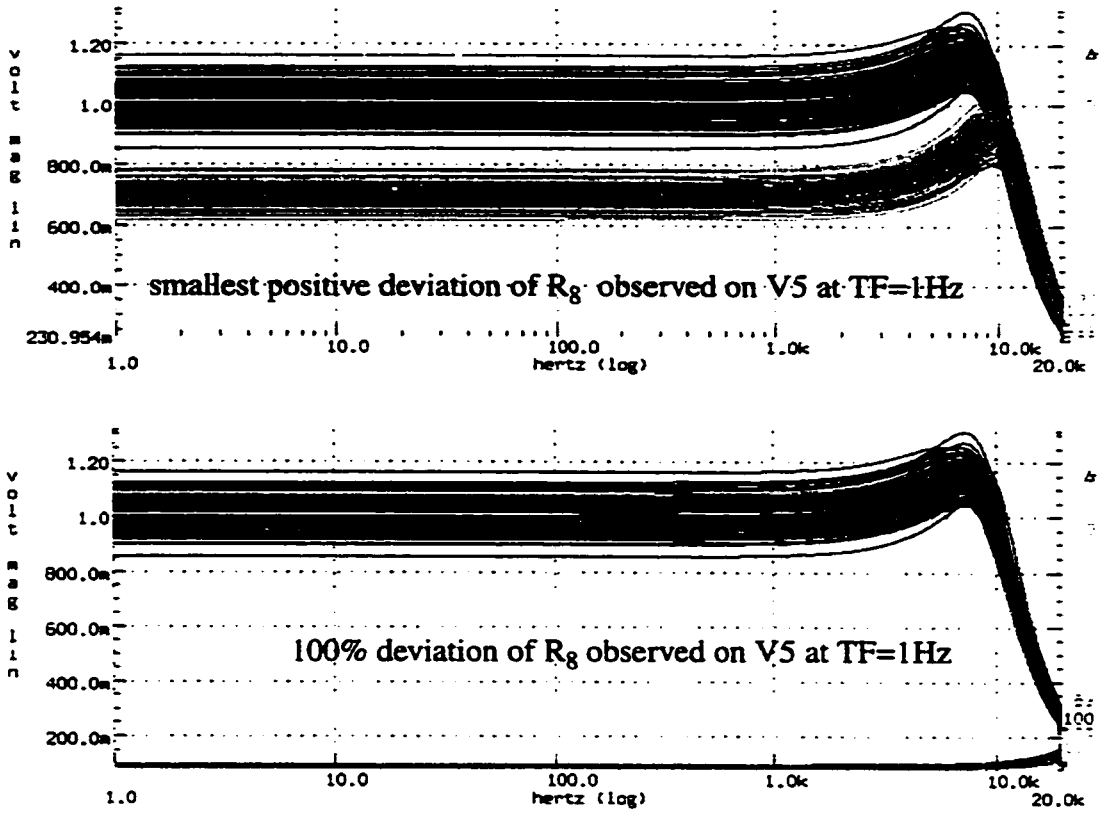
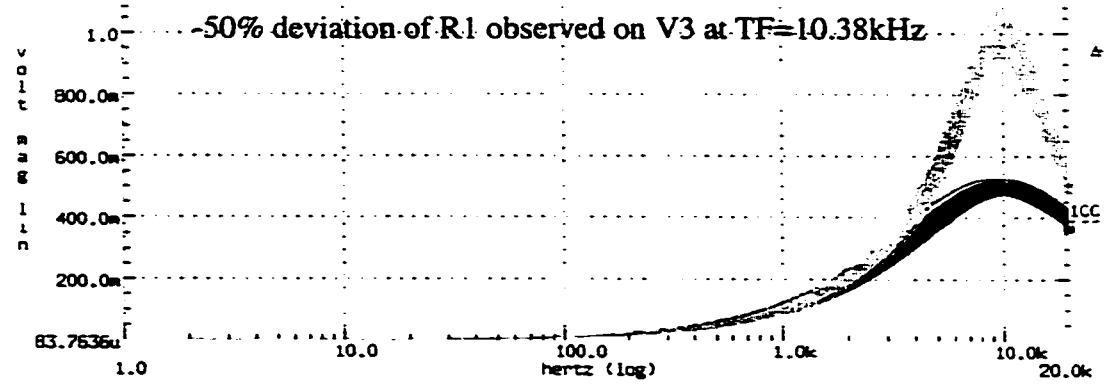


Figure 3.11: Faulty and good response families under faulty R_8 and the other parameters varied randomly.



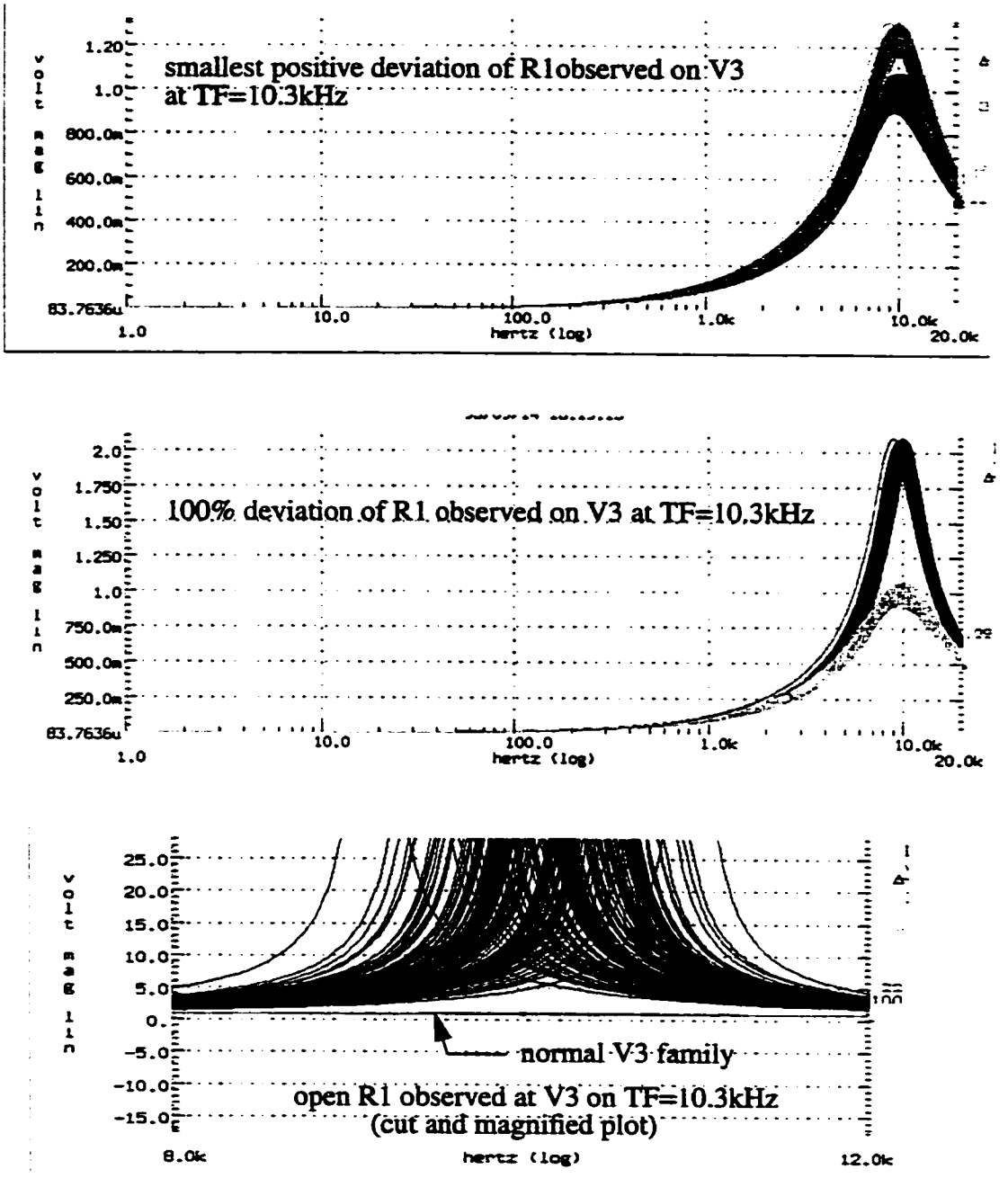


Figure 3.12: Faulty and good family responses under a faulty parameter R_1 and other random parameter variations

3.5 Conclusions

In this paper, we proposed a novel multifrequency test generation for detecting parametric and catastrophic failures in linear analog circuits. The test generation problem was formulated as a series of optimization problems. The method generates a robust test set that detects faults under maximal masking effects due to variations of parameters in their tolerance boxes. The proposed approach was illustrated on two examples. The computed smallest (largest) observable positive (negative) parameter deviations were validated using HSpice simulations of the observed performances which were then compared with the computed normal range (Tk_min, Tk_max). The magnitude of the errors obtained from these comparisons were very close to the resolution of the test equipment, thus confirming the accuracy of our method on these examples. Besides an adequate selection of test frequencies, it was shown that a complete solution to test generation problem needs an appropriate selection of performance and test point spaces.

In our future work, we aim at improving the approach to guarantee that the global optimum is always found regardless of properties of the performance functions. Also, we are elaborating a technique that allows to detect parameter variations that are difficult to observe (e.g., faulty deviations of R1 in the selective biquadratic filter).

References

[1] Semiconductor Industry Technology Workshop Conclusions, Semiconductor Industry Association, 1993.

[2] L. Milor and V. Visvanathan. "Detection of catastrophic faults in analog integrated circuits," IEEE trans. Computer-Aided Design, vol.8, pp.114-130, Feb. 1989.

[3] L. Milor and A. Sangiovanni-Vincentelli, "Optimal Test Set Design for Analog Circuits", Proc. ICCAD, pp. 294-297, 1990.

[4] L. Milor and A. Sangiovanni-Vincentelli, "Minimizing Production Test Time to Detect Faults in Analog Circuits", IEEE trans. Computer-Aided Design, vol.13, No.6, pp.796-813, June 1994.

[5] Shen-Jen Tsai, "Test Vector Generation for Linear Analog Devices", Proc. IEEE International Test Conference, pp. 592-597, 1990.

[6] N. Ben Hamida and B. Kaminska, "Analog Circuit Testing Based on Sensitivity Computation and New Circuit Modeling", Proc. IEEE International Test Conference, pp. 652-661, 1993.

[7] N. Nagi, A. Chatterjee, A. Balivada and J. Abraham, "Fault-based Automatic Test Generator for Linear Analog Circuits", Proc. of ICCAD, pp. 88-91, 1993.

[8] M. Slamani and B. Kaminska, "An Integrated Approach for Analog Circuit Testing with Minimum Number of Detected Parameters", Proc. IEEE International Test Conference, pp. 631-649, 1994.

[9] G. Devarayanadurg and M. Soma, "Analytical Fault Modeling and Static Test Generation for Analog ICs", Proc. of ICCAD, pp. 44-47, 1994.

[10] Optimization Toolbox User's Guide, December 1992, MathWorks, Inc.

[11] K. Schittowski, "NLQPL: A Fortran-subroutine Solving Constrained Nonlinear Programming Problems", Operations Research, Vol. 5, 485-500, 1985.

[12] J. W. Bandler, P. C. Liu, and J. H. K. Chen, "Worst Case Network Tolerance Optimization", IEEE Trans. Microwave Theory Tech., vol. MTT-23 pp. 630-641, Aug. 1975.

[13] H. Tromp, "The generalized Tolerance Problem and Worst Case Search", Proc. IEEE Conf. Computer-Aided-Design of Electronic and Microwave Circuits and Systems, pp. 72-77, July 1977.

Chapitre 4

Algorithmes de recherche de solution globale des problèmes de génération de tests et d'analyse de tolérance, basés sur la technique CLP

RÉSUMÉ

La méthode de programmation quadratique séquentielle (SQP) utilisée dans le chapitre précédent pour résoudre les problèmes d'optimisation formulés, fait partie des algorithmes standards qui ont tendance à trouver les optimums locaux d'une fonction qui sont situés le plus près du point de démarrage (*starting point, guess point*). En effet, puisqu'elles utilisent et traitent des informations locales (dérivée de la fonction objective en un point), ces méthodes ne peuvent pas garantir que la solution trouvée est globale. Il en résulte que l'application des méthodes d'optimisation standards au problème de génération de tests, ne peut pas garantir la qualité des tests générés. En outre, ces techniques ne sont pas automatiques et dépendent de plusieurs paramètres qui doivent être choisis d'une façon appropriée par l'usager. Par conséquent, afin de garantir une solution globale aux problèmes d'optimisation en question, et d'assurer ainsi la qualité des tests qui en découlent, il est nécessaire de mettre en oeuvre une méthode permettant d'atteindre cet objectif. Pour ce faire, nous avons élaboré une

méthode basée sur la programmation logique à contraintes (*Constraint Logic Programming: CLP*). Cette méthode fait l'objet du présent chapitre. Elle permet de résoudre la série des problèmes d'optimisation comme une série de problèmes de satisfaction à contraintes (*Constraint Satisfaction Problems: CSP*). La méthode proposée dans ce chapitre garantit que la solution trouvée est effectivement globale. Cela provient des propriétés de l'arithmétique relationnelle d'intervalles sur laquelle est fondée la technique CLP. Parmi ces propriétés figurent la propriété d'inclusion des intervalles, l'exactitude de processus de rétrécissement de ces derniers, ainsi que l'arrondissement des résultats vers l'extérieur.

L'efficacité de la méthode a été illustrée tout d'abord sur plusieurs fonctions non-linéaires reconnues comme difficiles. Ensuite, elle a été appliquée à plusieurs circuits électriques dans le contexte de génération de tests.

Bien que le temps CPU obtenus dans l'exécution des algorithmes de la méthode proposée est relativement raisonnable, nous proposons également dans ce chapitre des techniques permettant d'accélérer davantage ces algorithmes.

Enfin, il est important de mentionner que la qualité du procédé de fabrication ainsi que la résolution de l'appareil de mesure peuvent, dans certains cas, influencer largement la qualité des tests générés. Cette question sera également traitée dans le présent chapitre.

Ce travail a été publié en partie dans "*15th IEEE VLSI Test Symposium*" [3], et soumis pour publication en sa totalité à la revue "*IEEE Transactions on Computer-Aided Design*" [4]. Il est décrit en détail dans ce qui suit.

Worst-Case Tolerance Analysis and CLP-based Multifrequency Test Generation for Analog Circuits

A. Abderrahman[1], **E. Cerny**[2], **B. Kaminska**[1]

[1]Département de Génie Électrique et d'Informatique
Ecole Polytechnique de Montréal
C.P 6079, Succ. Centre-Ville;Montreal,Qc,Canada,H3C 3A7
[2]Département d'Informatique et de Recherche Opérationnelle
Université de Montréal
C.P 6128, Succ. Centre-Ville; Montréal, Qc, Canada H3C 3J7

Abstract

In our previous work we elaborated a multifrequency test generation method (TPG) for detecting parametric and catastrophic faults in linear analog circuits. The method is formulated as a series of optimization problems which were originally solved by Sequential Quadratic Programming (SQP), a non-linear programming method available in MATLAB. Such standard optimization methods are based on and process local information and consequently cannot guarantee a global optimum. Furthermore, they are not automatic and depend on various parameters which must be selected by an experienced user. In this paper, we propose a method based on Constraint Logic Programming(CLP) that solves these optimization problems in TPG as a series of Constraint Satisfaction Problems (CSPs). Our TPG method is fully automatic and provides tight and guaranteed bounds on the global optima of a multivari-

able nonlinear function. This quality of tightness and correctness of the bounds stems from the inclusion properties of intervals, outward rounding, and the correctness of narrowing of CLP implemented using Relational Interval Arithmetic (RIA). The TPG method was implemented in CLP(BNR) Prolog. First, we illustrate the effectiveness of our approach on a number of nonlinear functions known to be difficult, and then we apply it to a realistic electronic circuit in the context of TPG. The results are compared with those obtained by SQP. Although there is a good match between the results of the two methods, there is one case where SQP falls into a local minimum. This means that for some circuits it could lead to a wrong test selection. In addition, our method executes automatically in a matter of minutes, while the use of SQP requires much user interaction and may take more than a week of hard work. Moreover, we show that the use of monotonicity test and mean-value form can further speed-up the proposed approach. We also show that the tester resolution and the quality of the manufacturing process may strongly affect the parameter detectability, i.e., the test quality. Our method may also be used to solve worst-case tolerance analysis of circuit performances problem which plays an important role in design verification. This problem is efficiently tackled here as part of the test generation.

4.1 Introduction

Analog circuits are traditionally tested by verifying their function, which is known to be costly. Indeed, the estimated cost of analog testing may represent 30% of the total manufacturing cost [1]. Furthermore, in some applications, 95% of total test-

ing costs are spent in testing of analog circuits and 5% in testing of digital circuits, while 95% of the chip are digital and 5% are analog [2]. To minimize the test time and thus the cost of production testing of analog circuits, test generation techniques based on fault models are required. The difficulty of testing analog circuits stems from the fact that, unlike in digital circuits, the physical quantities that characterize them vary over time in a continuous range. This implies a continuum of possible defects. Consequently, there is a lack of adequate fault models, since the output values of analog circuits cannot be considered as either high or low levels as in the digital world where a large class of defects can be modeled by stuck-at-0/1faults.

In general, faults in analog circuits can be classified into hard and parametric faults. Hard faults are caused by catastrophic variations in parameter values such shorts and opens, and usually induce a complete loss of correct functionality. Parametric faults are caused by an abnormal deviation of parameter values and result in altered performance. Both types of faults have to be detected by a test set. Milor et al. [3] reported on a test generation algorithm for detecting catastrophic faults under normal parameter variations. In [4,5] an approach based on a statistical process fluctuation model was derived to select a subset of circuit specifications that detect parametric faults and minimize the production testing time. Test generation is formulated in [6] as a quadratic programming problem. This approach was developed for parametric faults and it determines an input stimulus x(t) that maximizes the quadratic difference of responses from the good and the faulty circuits with other parameters at their nominal values. A test generation approach for hard and parametric faults based on sensitivity analysis and tolerance computation was exposed in [7]. In this approach the worst case performance was expressed in terms of sensitivity and parameter toler-

ance, however, frequency analysis was not considered, and the model was a linearization obtained from first order partial derivatives. In [8], which is based on [7] and [10], the authors developed an automated sensitivity tool (LIMSoft) that allows adjoint network-based sensitivity computation and analysis for designing fault-resistant circuits and for generating test vectors for parametric and catastrophic faults under normal parameter variations.

The method presented in [9] is founded on a fault-model and sensitivity. For a given fault-list, perturbation of sensitivity with respect to frequency is used to find the direction toward the best test frequency. In [10] the authors derived a multifrequency test generation technique based upon testability analysis and a fault observability concept. The test frequencies selected are those where the output performance sensitivity is maximum with respect to faulty component deviation. In these approaches [9-10] the masking effects due to variations of the fault-free components in the tolerance box are not considered and the test frequencies may be not optimal. A DC test generation technique for catastrophic faults was developed in [11]. This technique is fault-based and test generation is formulated as an optimization problem including the effects of parameter variations.

Any robust test set has to detect parametric and hard faults under maximal masking effects due to normal variations of parameters. Indeed, only in this case the quality of a test set may be correctly measured and guaranteed. In our previous work [12], we proposed a novel test generation approach for detecting hard and parametric faults in analog circuits. This method is based on multifrequency testing which is, in general, more suited for subtle parameter variations than DC testing. Given a set of perfor-

mances and a frequency range, it selects the test frequencies that maximize the observability of a parameter deviation on a circuit performance under the worst masking effects of normal variations of the other parameters. As in [11], the test generation is formulated as a series of optimization problems. To solve them, a nonlinear programming method SQP (Sequential Quadratic Programming) available in MATLAB was used. Such standard optimization algorithms tend to find the local minima nearest the starting point. Indeed, since they are based on and process local information (such as the derivatives of the objective function at a point which may not be reliable), they cannot guarantee finding the global optimum.

In this paper, we propose a method based on Constraint Logic Programming(CLP) that solves the optimization problem as a series of Constraint Satisfaction Problems (CSPs) and provides tight and guaranteed bounds on the global optima of a nonlinear function.

The paper is organized as follows. In the next Section, the problem formulation of the test generation of [12] is reported here for convenience. CLP and connected techniques are outlined in Section 3. The proposed CLP-based method for solving the optimization problems is described in Section 4. Experimental results are reported in Section 5. The impact of tester resolution and the quality of the manufacturing process on test quality are discussed in Section 6. Acceleration techniques are described in Section 7 and conclusions are drawn in Section 8.

4.2 Problem formulation

4.2.1 Objectives

The goal is to generate the smallest set of robust tests that maximize the observability of a parameter deviation on a circuit performance under the worst masking effects of normal variations of the other parameters. This means that, for each parameter, we determine a) Its smallest absolute possible deviation outside which its detectability can be guaranteed under any variation of the other parameters within their tolerances, and b) the frequency of the input signal (the test) which guarantees this detection.

Let T be the performance of interest of the circuit under test, and $T_{max}(f)$ and $T_{min}(f)$ the extreme values of T under the normal parameter variations that determine the acceptance range of T at frequency f. A circuit is declared faulty if a test produces a performance outside its acceptance range $[T_{min}(f), T_{max}(f)]$. In this case, the parameter deviation associated with the fault is said to be observable. A performance may be selected as a test performance if it detects the smallest absolute minimum observable deviation of at least one parameter x_i (i=1,...,m) of a circuit at some frequency f.

4.2.2 Problem Formulation

Let $T(f, \mathbf{x}) = T(f, x_1, \ldots, x_m)$ be the performance to observe, as function of frequency f in $[f_{min}, f_{max}]$, $\boldsymbol{x} = [x_1 x_2 \ldots x_m]^T$ the vector of parameters of the circuit, and $\boldsymbol{x}_n = [x_{1n} x_{2n} \ldots x_{mn}]^T$ the nominal value of x. Let the normal tolerance range of a

parameter x, be the interval $X_i = [x_{il}, x_{iu}]$, i=1,....,m, and the total possible range of values x, can take be from the interval $[\underline{x}_{il}, \bar{x}_{iu}]$, $\underline{x}_{il} \le x_{il}$ and $\bar{x}_{iu} \ge x_{iu}$. Denote $\boldsymbol{x}_l = [x_{1l}x_{2l}...x_{ml}]^T$ and $\boldsymbol{x}_u = [x_{1u}x_{2u}...x_{mu}]^T$. Under normal circumstances we thus have $\boldsymbol{x}_l \le \boldsymbol{x} \le \boldsymbol{x}_u$, and $\boldsymbol{x}_l \le \boldsymbol{x}_n \le \boldsymbol{x}_u$.

4.2.2.1 Worst-case tolerance analysis

Let $T_{min}(f, \mathbf{x})$ and $T_{max}(f, \mathbf{x})$ be the extreme values of T at frequency f under the variation of $\boldsymbol{x} \in [\boldsymbol{x}_l, \boldsymbol{x}_u]$. These extreme values can be obtained as the solution of the following optimization problems.

For a given frequency f,

$$T_{max}(f) = \underset{\boldsymbol{x}}{MAX} \; T(f, \boldsymbol{x}) \tag{1}$$

subject to $\boldsymbol{x}_l \le \boldsymbol{x} \le \boldsymbol{x}_u$

and

$$T_{min}(f) = \underset{\boldsymbol{x}}{MIN} \; T(f, \boldsymbol{x}) \tag{2}$$

subject to $\boldsymbol{x}_l \le \boldsymbol{x} \le \boldsymbol{x}_u$

The envelope as delimited by $T_{max}(f)$ and $T_{min}(f)$ constitutes the acceptance range of T.

The optimization problems (1) and (2) represent the formulation of the analysis of worst-case tolerances of circuit performances. In this paper we propose a new method for tackling this problem as part of the test generation problem. However,

worst-case tolerances play an important role in circuit design verification. Fluctuations in the manufacturing process cause circuit parameters to vary about their nominal values. As a result, the performances of fabricated circuits will vary about their nominal values too. Some circuits may not meet their performance specifications, thus causing yield loss. A circuit design should be optimized in such a way that the greatest possible number of manufactured circuits perform satisfactorily.

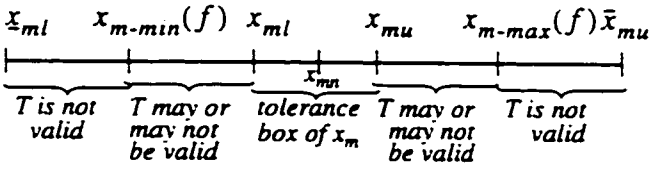
The difficulty of the worst-case tolerance analysis problem stems from the fact that it is a nonlinear, multivariate global optimization problem. There are mainly three approaches to deal with this problem. One approach is based on a Monte-Carlo method which is time-consuming and does not guarantee tight bounds. Non-linear programming methods are semi-automatic and cannot guarantee global optima. Interval arithmetic is by far superior with regards to computation time and accuracy. Several interval methods [20],[22], [23],[24],[25] can be seen as modifications and improvements to the algorithm proposed in [27]. This algorithm is based upon the branch and bound principle. It works by splitting up the initial box X into subboxes. The search of global minimum is done in these subboxes. at each iteration the search continues in the subbox (branching principle) with the smallest lower bound (bounding principle). The splitting process continues until the desired accuracy is reached. The algorithm was later improved by using the monotonicity test [20], yet it is quite time consuming.

4.2.2.2 Minimum observable parameter deviation [12]

Without loss of generality, let x_m be the parameter whose changes we wish to observe at performance T. Let $x_{m\text{-}min}(f) \in [\underline{x}_{ml}, x_{ml}]$ (resp. $x_{m\text{-}max}(f) \in [x_{mu}, \bar{x}_{mu}]$) be the smallest (resp. largest) value of x_m such that T(f) is outside of the interval $[T_{min}(f), T_{max}(f)]$ for all x_m in $[\underline{x}_{ml}, x_{m\text{-}min}(f)]$ (resp., for all x_m in $[x_{m\text{-}max}(f), \bar{x}_{mu}]$.

Graphically, we can visualize the various intervals of values of x_m as follows:

Figure 4.1 Total possible range of a parameter x_m under normal variations of x_i, $i \neq m$.



Given some frequency f, the objective is to determine the values $x_{m\text{-}min}(f)$ (resp. $x_{m\text{-}max}(f)$), regardless the values of the other x_i, $i \neq m$, within their respective normal intervals $[x_{il}, x_{iu}]$. As a result, this gives us a limit on the deviation of x_m outside its normal tolerance box, such that if x_m is smaller than $x_{m\text{-}min}(f)$ (larger than $x_{m\text{-}max}(f)$), then this (faulty) deviation is guaranteed to be detected at f (i.e., T would be outside the range $[T_{min}(f), T_{max}(f)]$) independently of the other parameter values within their normal variation intervals. Furthermore, we wish to find the frequency under which we can detect the largest value $x^*_{m\text{-}max}$ of $x_{m\text{-}min}(f)$ (resp. the smallest value $x^*_{m\text{-}min}$ of $x_{m\text{-}max}(f)$), i.e., the frequency which makes a faulty deviation of x_m

most observable. This problem can be formulated as a max-min (resp. min-max) optimization problem as follows:

Let $\mathbf{x}_m$ be the parameter vector $x_m = [x_1 x_2 \ldots x_{m-1}]^T$, and let δ be the resolution of the test equipment with respect to T. It follows that

$$x^*_{m\text{-}min} = \underset{f}{MIN} \; \underset{x_m}{MAX} \; (x_m) \tag{3}$$

subject to

$$T_{min}(f) - \delta \leq T(f, x) \leq T_{max}(f) + \delta$$

$$x_{il} \leq x_i \leq x_{iu} \quad i = 1, \ldots, m-1$$

$$x_{mu} \leq x_m \leq \tilde{x}_{mu}$$

$$f_{min} \leq f \leq f_{ma}$$

and

$$x^*_{m\text{-}max} = \underset{f}{MAX} \; \underset{x_m}{MIN} \; (x_m) \tag{4}$$

subject to

$$T_{min}(f) - \delta \leq T(f, x) \leq T_{max}(f) + \delta$$

$$x_{il} \leq x_i \leq x_{iu} \quad i = 1, \ldots, m-1$$

$$x_{ml} \leq x_m \leq x_{ml} \quad \text{and} \quad f_{min} \leq f \leq f_{ma}$$

In general, these are complex non-linear optimization problems. We discretize the frequency in the interval of interest $[f_{min}, f_{max}]$ into a set $\{f_j\}$ of p frequencies and solve the remaining MIN (resp. MAX) problem at each f_j. In [12] we used SQP non-linear programming method available in MATLAB to solve these problems, however,

it may take more than a week of hard work to set up the appropriate initial conditions and to gain confidence about the correctness of the results. In Section 4, we propose a new method that transforms these problems into a series of Constraint Satisfaction Problems (CSPs) that are efficiently solved using Relational Interval Arithmetic (RIA). The method is fully automatic and finds guaranteed solutions. Before describing this method, we introduce in the next Section Constraint Logic Programming (CLP) and its connected techniques.

4.3 Constraint Satisfaction Problems

4.3.1 Constraint Logic Programming

Constraint Logic Programming (CLP) is a combination of constraints solving and logic programming. This combination provides a unified framework for modeling and solving Constraint Satisfaction problems (CSPs). Although a relatively new field, CLP has been used in many application areas [13]. The CLP we used is based on Relational Interval Arithmetic (RIA) which draws its basic concepts from functional interval arithmetic.

4.3.1.1 Functional Interval Arithmetic

The principles of functional arithmetic were first explored by Moore [14] to deal with the incorrect behaviors of finite precision arithmetic. An interval $A = [x_l, x_u]$ is the set of real numbers x such that $x_l \leq x \leq x_u$. The width of an interval A is defined by $w(A) = x_u - x_l$. An interval A is called degenerate if its width is null. So, a real number

can be processed as degenerate intervals (in fact as the smallest floating point interval containing the number). For $A = [x_l, x_u]$ and $B = [y_l, y_u]$, functional interval operations are defined by: $A + B = [x_l + y_l, x_u + y_u]$, $A - B = [x_l - y_u, x_u - y_l]$, $A*B = [\min(x_l*y_l, x_l*y_u, x_u*y_l, x_u*y_u), \max(x_l*y_l, x_l*y_u, x_u*y_l, x_u*y_u)]$, $A/B = [x_l, x_u]*[1/y_u, 1/y_l]$. An important operation is the intersection of two intervals which is defined by: $A \wedge B = [\max(x_l, y_l), \min(x_u, y_u)]$. A rational interval function $F(X_1, ..., X_m)$ can be derived from the rational function $f(\mathbf{x}) = f(x_1, ...x_m)$ by replacing the real variables $x_1, ..., x_m$ by intervals $X_1, ..., X_m$ and by replacing the arithmetic operations by the equivalent interval functional operations. F is called the natural interval extension of f.

Range of a function

Let $X = (X_1, ..., X_m)$ be a box (an interval vector), the useful interval function called the united extension is defined by $f(X) = \{f(x_1, ...x_m) \mid x_i \in X_i, i=1, m\}$.

The power of the functional interval arithmetic derives from the following theorem due to Moore [14]:

Theorem 1: If F (X) is an inclusion monotonic interval extension of f (x), then

$$f(X) \subseteq F(X)$$

that is, the interval extension F (X) contains the range of $f(x_1, ...x_m)$ for all $x_i \in X_i$, i=1, m.

Let's illustrate this theorem by a simple example: $f(x) = x^2$ over X=[-1, 2]. It can be easily seen that the united extension f (X) = f ([-1, 2]) = [0, 4] since the range of f (x) over the domain [-1, 2] is [0, 4]. On the other hand, the interval function F (X)

= F([-1, 2]) = X*X = [-1, 2]*[-1, 2] = [-2, 4]. Hence $f(X) \subset F(X)$, that is F(X) represents wider interval than f (X). This is in general true except the case described by the following theorem [15].

Theorem 2: If F (X) is any natural extension of a rational function in which each variable occurs not more than once and to the first power, then F (X) = f (X) provided no division by interval containing zero occurs.

For example, in f (x) = x_1*x_2*x_3 the variables x_1, x_2 and x_3 occur once and Theorem 2 can be applied.

From Theorem1, we can deduce that if the interval functional operations are implemented as described above any computed interval function is an inclusion. As a result it is guaranteed to include all real solutions. That is, we can find infallible bounds on the range of a function f(X) over the box X just by computing the interval extension F(X). Unfortunately, these bounds will not be tight enough, in particular, when the box is large. Consequently, one of the most important problems to deal with is to find tighter bounds on f(X) with an acceptable computation effort.

4.3.1.2 Relational Interval Arithmetic (RIA)

Relational arithmetic on real intervals was proposed by John Cleary in [16] to supply a relational model of arithmetic operations on intervals in Prolog. RIA and Functional Interval Arithmetic use the same basic concepts of intervals: inclusion properties and outward rounding. The main differences are relations versus functions, and the use of narrowing operators by RIA. Only some of the basic relational opera-

tions are described here [17]. The equality relation reduces to interval intersection. That is if A = [x_l, x_u] and B = [y_l, y_u] are constrained by equality relation A == B, then both A and B are narrowed to the interval A^B = [max(x_l, y_l), min(x_u, y_u)]. This concept of equality is different from equality in functional interval arithmetic where two intervals are equal *iff* their bounds are the same. The inequality relations may also narrow intervals. If $A \leq B$, the resulting intervals are computed by intersecting A with [$-\infty$, y_u] and B with [x_l, $+\infty$]. For example, if A = [2, 4], B = [1, 6] and $A \leq B$ is evaluated, then B is narrowed to [2, 6] and A is not narrowed. The most complex operation is multiplication for which a number of cases should be distinguished.

One of the advantages of relational arithmetic is illustrated by the equality relation R*R == A*A + B*B, from which we can calculate not only R from A and B, but also A from R and B, and B from R and A, depending on the unknown variable.

Narrowing algorithm [18]: The main properties of the narrowing algorithm are Contractance (the narrowed intervals are smaller than the initial intervals), Correctness (every real solution lies in the narrowed intervals), Monotonicity (narrowing preserves inclusion) and Idempotence (the narrowed intervals have to be computed but once).

4.3.1.3 Constraint Satisfaction Problems

We now introduce some definitions related to Constraint Satisfaction Problems (CSPs) [19].

Definition 1: a CSP

A CSP S = (**x**, X, C) is defined by

- a set of numeric variables $\mathbf{x} = \{x_1, ...,x_m\}$

- a set of domains $X = \{X_1, ..., X_m\}$ where X_i, a set of numeric values, the domain associated with the variable x_i.

- a set of constraints $C = \{C_1, ..., C_p\}$ where a constraint C_i is defined by any numeric relation linking a set of variables.

Definition 2: solution to a CSP

A solution to a CSP S = (**x**, X, C) is an instantiation of the variables of **x** for which both the inclusion in the associated domains and all the constraints of C are satisfied.

Definition 3: global consistency of a CSP

A CSP S is globally consistent *iff* $\forall x_i \in \boldsymbol{x}, \forall a \in D_i, x_i = a$belongs to a solution to S.

The search for global consistency is an incomputable problem over real numbers, but in finite precision it becomes an NP-hard problem because enumeration is possible.

Definition 4: an empty CSP

A CSP S is empty *iff* $\exists D_i \in D$, such that D_i is the empty set.

That is, a CSP S is inconsistent *if* it admits no solutions. In the next section we proceed to describe our new method for solving the optimization problems.

4.4 Proposed CLP-based approach for solving optimization problems

In this section we will describe our method to solve the optimization problems (1), (2), (3) and (4) presented in Section 2.

4.4.1 Worst-case tolerance analysis

Finding the global optima (max and min) of a multivariate function over an n-dimensional rectangle (box), is equivalent to bounding its range [20].

Consider, for a given frequency f, the performance T (**x**) = T (f, x_1, ... x_m). Let $x_i \in X_i = [x_{il}, x_{iu}]$, and X = ($X_1$, ..., X_m) so that $x \in X$. Let T (X) be an inclusion function of T (**x**). In the following, we describe an algorithm for computing the range T (X) = [T_{min},T_{max}] over the box X, i.e., the global minimum T_{min} and maximum T_{max} of T(X). Note that for polynomial functions, the nested form (also called Horn's scheme) of an inclusion function provides bounds that are usually less wide than the simple sum of powers. For example, let $f(x) = 2x_1^2 - 1.06x_1^4 + \frac{1}{6}x_1^6 - x_1x_2 + x_2^2$, the inclusion function $F_1(X) = 2X_1^2 - 1.06X_1^4 + \frac{1}{6}X_1^6 - X_1X_2 + X_2^2$ over the box X = ([-2, 4],[-2, 4]) gives the range F_1 = [-287.36, 738.666]. On the other hand, the nested form

$F_2(X) = X_1^2\left(2 - X_1^2\left(1.06 - \frac{1}{6}X_1^2\right)\right) - X_1X_2 + X_2^2$ provides a much narrower range F_2 = [-255.36, 467.3066]. As a result, an adequate selection of an inclusion function may help to reduce the computation effort. From now on, we shall assume that an appropriate inclusion function of the function under consideration has been selected.

At the beginning of the algorithm, T(X) is a free variable and the variables x_i, i=1,m, are constrained to be within their tolerance interval $X_i = [x_{il}, x_{iu}]$. Note that an unbound variable is equal to the universal interval which ideally varies from minus infinity to plus infinity, but it depends on the specific hardware implementation of floating point numbers. Then, the first step is to reduce this large search space of T(X) by evaluating the equality relation:

$$T = = T(f, X_1, \ldots, X_m) \qquad (5)$$

whose effect is to bound T within $D_T = [T_l, T_b]$ thanks to the narrowing algorithm. Let $\mathbf{x}_n = (x_{1n}, \ldots, x_{mn})$ be the nominal parameter vector with $x_{in} = \frac{x_{il} + x_{iu}}{2}$, i=1, ...m, and $T_n = T(\mathbf{x}_n)$ be the nominal value of $T(\mathbf{x})$. Then, we can divide the domain D_T and reduce further the search space by searching T_{max} in $D_{max} = [T_n, T_b]$ and T_{min} in $D_{min} = [T_l, T_n]$.

4.4.1.1 The basic approach

Instead of searching for global consistency of a CSP which is an NP-hard problem and to reduce the search space, our method proceeds by eliminating those parts of the domain where the CSP is inconsistent. To find T_{max} (resp. T_{min}), the idea is to constrain T by recursive binary splitting of the domain D_{max}(resp. D_{min}), to within the

restricted domain D_R (resp. D_L) which is equal to the right part (resp. the left part) of the splitted current domain. As a result T is constrained to a sequence of domains D_R (resp. D_L) which progressively become tighter and tighter. The propagation of these constraints through the evaluation of the equality relation (5) causes the intervals X_i, i=1,...., m, to be narrowed progressively more or less depending on the width $w(D_R)$ (resp. $w(D_L)$) and their sensibility to the variation of T. In particular, when D_R (resp. D_L) is reduced to a point interval, the intervals X_i may become near points or empty. If the constraint system detects an inconsistency, that is an empty interval X_i, we can affirm that the solution is not included in D_R (resp. D_L). As a result, the domain D_R (resp. D_L) is eliminated from the search space, and the search continues in the remainder of the current domain D_L (resp. D_R). The process of splitting, narrowing, and detecting inconsistencies continues until the desired accuracy is reached. However, in some cases this process provides an acceptable bounds, but in other cases it can not lead alone to bounds that are tight enough. One way to get sharp bounds on T_{max} and T_{min} is to use in each subdomain the predicate ***solve*** provided in CLP(BNR) Prolog [21]. ***solve*** is an algorithm for searching subintervals that satisfy the constraints imposed by the problem. When using ***solve***, failure implies that no solution exist in the initial ranges, whereas success only indicates that there may be solutions in one or more of the final ranges returned through Prolog backtracking.

Our goal here is to eliminate any part (of the initial domain) of T in which there are no solutions. Therefore, conjugated with the narrowing process described above, ***solve*** is used here to detect system inconsistency, i.e., that no solution exists in the current part of the domain.

4.4.1.2 Description of the Optimization Algorithm

We first seek the value T_{max} and describe the corresponding algorithm. We assume that the nominal value T_n of T is known. By evaluating the equality relation (5) with T in $D_T = [T_n, \quad +\infty]$ and $X_i = [x_{il}, x_{iu}]$, i=1, ..., m, D_T becomes bounded, i.e., $D_T=[T_n,T_b]$. As mentioned above, the narrowing process and the detection of system inconsistencies are the heart of the proposed method. This task is achieved by the procedure *Is_consistent*. When called, it evaluates the equality relation (5) which may cause the intervals X_i, i=1, ..., m, to be narrowed, and then applies to them the predicate ***solve***(X_i) by recursive calls. ***solve***(X_i) attempts to split the interval X_i to generate the disjoint solutions to the constraints imposed on T and on intervals X_i, i=1, ..., m. If the current constraints are not satisfied, then there are no solutions in the current domain of T. It is important to note that the combined use of simple variables and combinations of variables on which ***solve*** is applied may speed up the detection of inconsistency.

The algorithm starts its search by constraining T to the degenerate interval $D_T = [T_b, T_b]$ and calls *Is_consistent* to check if *there are solutions*. If the procedure *succeeds*, then the maximum T_{max} is equal to T_b, and the algorithm is stopped. On the other hand, if the procedure fails, i.e, *there are no solutions*, the splitting process starts and the domain $D_T = [T_n, T_b]$ is divided into two equal parts left and right, denoted by $D_L = [T_n, Mid]$ and $D_R = [Mid, T_b]$, respectively, with $Mid = (T_n + T_b)/2$. Next, we constrain the domain D_T to $D_T = D_R$ and call the procedure *Is_consistent*. If it fails the domain D_R is eliminated, and the search continues in the domain D_L which is divided into two equal parts and the process is repeated. If it succeeds, D_L is pushed on a

LIFO stack L (because D_L may contain the solution and may thus be processed later). Then, the search continues in the domain D_R which is divided into two equal parts, and the process is again repeated. The stack L is used here for the sake of clarity. In fact, since we perform dichotomic splitting, we can recompute the bounds of its elements when needed. Here we should emphasize the fact that the solution T_{max} is approached from the right by progressive elimination of those parts of the initial domain D_T in which there are no solutions, i.e., the current constraints are not satisfied.

The process continues until the termination criterion is reached, that is the width $w(D_T) \leq \varepsilon$ (where ε is a small value > 0). We distinguish two cases here when the procedure *Is_consistent* is called: 1) If it succeeds then T_{max} is found and is equal to D_T, and the algorithm is stopped. 2) If it fails, then constrain D_T to the first element of L, and call the procedure *Is_consistent*. While it fails constrain D_T to the next element of L. If it succeeds and $w(D_T) > \varepsilon$ divide D_T into two equal parts and continue the process as described above. Otherwise, if it succeeds and $w(D_T) \leq \varepsilon$, then T_{max} is found and is equal to D_T, and the algorithm is stopped.

Algorithm 1

Begin

get {T, f, X, ε} *\ T is a performance to observe at a given frequency f, X = (X_1,..... X_m) is the interval parameter vector, with $X_i = [x_{il}, x_{iu}]$, i=1, ..., m, and ε is the desired accuracy.

1- Initialize the stack L to empty, calculate the nominal value T_n, initialize $D_T = [T_n, \quad +\infty]$, and call the procedure *Evaluate* (D_T,T, f, X_1, ... X_m). This results in $D_T = [T_n, T_b]$;

2- Set $D_T = [T_b, T_b]$;

3- *Is_consistent* (D_T, T, f, X_1, ..., X_m);

4- If success then print "Tmax = T_b" and STOP. Else, set $D_T = [T_n, T_b]$;

5- Divide D_T into two halves D_L and D_R;

6- Constrain $D_T = D_R$;

7- *Is_consistent* (D_T, T, f, X_1, ..., X_m);

8- If success then push D_L on stack L. Go to 15;

9- If failure then discard D_R and set $D_T = D_L$. If $w(D_T) > \varepsilon$ then go to 5;

10- *Is_consistent* (D_T, T, f, X_1, ..., X_m);

11- If success then print "$T_{max} = D_T$" (since $w(D_T) = w(D_L) \leq \varepsilon$) and STOP;

12- If failure then set D_T = top(L), pop(L);

13- *Is_consistent* (D_T, T, f, X_1, ..., X_m);

14- If failure then go to 12;

15- If $w(D_T) \leq \varepsilon$ then "$T_{max}=D_T$" and STOP, else go to 5;

End.

To search for the global minimum T_{min}, the same algorithm described above is applied but in a symmetrical manner, that is, we now approach the solution from the left by progressive elimination of the parts of the domain $D_T=[T_l,T_n]$ in which there are no solutions.

4.4.1.3 Numerical Examples

The method outlined above was implemented in CLP(BNR) Prolog, and the following experiments were carried out on a SPARC-Station 10.

Example 1: Consider the so-called "three-hump-camel function" [22]:

$$f(x) = 2x_1^2 - 1.06x_1^4 + \frac{1}{6}x_1^6 - x_1x_2 + x_2^2$$

It has two local minima at approximately $(\pm 1.75, \pm 0.87)$, two saddle points at approximately $(\pm 1.07, \pm 0.535)$ and one global minimum at (0,0). All these points are contained in the box X=([-2,4],[-2,4]). The results shown in Table 4.1 were computed with the accuracy $\varepsilon = 10^{-4}$. Those obtained by the algorithm of [23] are also shown for comparison.

TABLE 4.1 Range of three-hump-camel function

range of f	Our algorithm	Algorithm in [23]
$[f_{min}, f_{max}]$	$[-2.2357\,10^{-5}, 455.3067]$	$[-1.8208\,10^{-14}, 457.8666]$
CPU time(s)	12.82	not available

From Table 4.1, we can observe that the range of f(**x**) obtained by our algorithm is narrower than that reported in [23], and it was obtained in a relatively short time.

Example 2: Consider now the "six hump camel-back function" [24]:

$$f(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$$

The domain is X = ([-2.5, 2.5], [-2.5, 2.5]). The optima shown in Table 4.2 were computed with the accuracy $\varepsilon = 10^{-4}$. Notice that the minimum of $f(x)$ computed by our method is similar to the one obtained by the algorithm in [23].

TABLE 4.2 Range of six-hump-camel function

Optimum	Our algorithm	Algorithm in [23]
f_{min}	-1.0320	-1.0316
f_{max}	161.8489	not available
Total CPU time (s)	69.8	not available

Example 3: In this example we consider the following function [23]:

$$f(x) = f_1(x_1)f_2(x_2)f_3(x_3)f_4(x_4)f_5(x_5) \text{ where}$$

$$f_1(x_1) = 0.01x_1(x_1 + 13)(x_1 - 15)$$

$$f_2(x_2) = 0.01(x_2 + 15)(x_2 + 1)(x_2 - 8)$$

$$f_3(x_3) = 0.01(x_3 + 9)(x_3 - 2)(x_3 - 9)$$

$$f_4(x_4) = 0.01(x_4 + 11)(x_4 + 5)(x_4 - 9)$$

$$f_5(x_5) = 0.01(x_5 + 9)(x_5 - 9)(x_5 - 10)$$

The domain is $X_1 = [8, 9]$, $X_2 = [-10, -9]$, $X_3 = [-5, -4]$, $X_4 = [3, 4]$, $X_5 = [-3, -2]$. The results were computed with an accuracy $\varepsilon = 10^{-6}$ and for comparison they are shown in Table 4.3 with those obtained by the algorithms in [23] and [25].

TABLE 4.3 Range of a composed function

Range of f	Our algorithm	Algorithm in [25]	Algorithm in [23]
$[f_{min}, f_{max}]$	[23067.372,24416.036]	[23067.370,24416.040]	[22110.018,25426.918]
CPU time(s)	88.43	not available	not available

From Table 4.3, we can observe that the range of f(x) computed by our algorithm is similar to that reported in [25], but is much narrower than the one calculated by the algorithm in [23].

We can also observe from Tables 4.1, 4.2, and 4.3 that the CPU time needed by our algorithm to find the global optima is reasonable, considering the difficult structure of these functions. Note that the important aspect of our method is that it finds a tight upper (lower) bound on the max (min) value of a function. This is essential for our TPG application.

4.4.2 Finding optima (max and min) of observable parameter values

In this section we describe an algorithm for computing the maximum (resp. minimum) observable value $x_{m_max}(f)$ (resp. $x_{m_min}(f)$) of a parameter x_m over the box $X^{(m)} = [X_1, X_2, \ldots, X_{m-1}]$ such that T is constrained to be within $[T_{min}(f) - \delta, T_{max}(f) + \delta]$, for a given frequency f. The expressions for these val-

ues are given as in (3) and (4) of Section 2, and only one of them is repeated here for convenience:

$$x_{m_max}(f) = \underset{x_m}{MAX} \quad (x_m) \tag{6}$$

subject to constraints as given in Section 2.

Since CLP (RIA) is founded on relational arithmetic, there is no need to derive the expression x_m = function (T, f, x_1, ..., x_{m-1}) which is, in general, very difficult (even impossible in some cases). Furthermore, since the optimization problem is viewed as a series of Constraint Satisfaction Problems (CSPs), we can easily add, modify and remove constraints to satisfy new goals. To solve problem (6) T should be constrained to be within $[T_{min}(f) - \delta, T_{max}(f) + \delta]$, while x_m becomes constrained to be within $X_m = [x_{mu}, \quad +\infty]$ (resp. $X_m = [0, x_{ml}]$). On the other hand, other parameters remain constrained to $X_i = [x_{il}, x_{iu}]$, i=1, ..., (m-1). Consequently, we obtain a new CSP S = (**x**, X, C) defined by the parameter vector **x** = (x_1, ..., x_m), the box X = (X_1, ..., X_m) and the set C of constraints constituted by the relation T == T(f, X_1, ..., X_m), with the domain constraint T = $[T_{min}(f) - \delta, T_{max}(f) + \delta]$.

To find the maximum observable value $x_{m_max}(f)$, it suffices to trim the parts of the domain $X_m = [x_{mu}, \quad +\infty]$ from the right until the algorithm detects that the current system of constraints is consistent. To guarantee that we find an upper bound on the true global maximum $x_{m_max}(f)$, it is important to stress the fact that the trimming operation should be performed progressively from the right of the domain X_m.

The actual Algorithm 2 given below proceeds as follows: Similarly as Algorithm 1, it splits the domain X_m into two equal parts, left and right, denoted by X_L and X_R, respectively. The procedure *Is_consistent* is used to check if the current set of constraints including the new constraint $X_m = X_R$ is consistent. If it fails then the domain X_R is discarded and the search continues in the left half X_L. It is divided into its left and right halves and the process is repeated. On the other hand, if *Is_consistent* succeeds, then X_L which may contain a solution is pushed on stack L_m for possible processing later. The search continues in X_R which is splitted into its left and right halves, and the process is repeated until a solution is found under similar criteria as in Algorithm 1. There are some specific considerations in the implementation of Algorithm 2, however. In some cases, the starting domain $X_m = [x_{mu}, \quad +\infty]$ of x_m remains very large even after the narrowing effects of the relation $T == T(f, X_1, \ldots, X_m)$ (see Step 1 of Algorithm 2). To overcome this problem and to reduce the computing time, X_m may be narrowed by considering the fact that an open resistance is usually modeled as 100 Mohms, and a shorted capacitance is modeled as 1F. Consequently X_m becomes $X_m = [x_{mu}, 10^8]$ for a resistance, and $X_m = [x_{mu}, 1]$ for a capacitance. Note that for each kind of parameter for which we search the max/min observable values, a termination criterion ε_m must be adequately chosen (resistances and capacitances cannot have the same ε_m).

Algorithm 2

Begin

get {T, f, X, δ, ε_m } **\ T is a performance to observe at a given frequency f, X = (X_1,..., X_m) is the domain vector, X_i = [x_{il}, x_{iu}], i=1, ..., m, δ is the tester resolution, and ε_m is the desired accuracy.

1- Initialize the stack L_u to empty, initialize X_m=[x_{mu}, $+\infty$], and call the procedure *Evaluate* (T, f, X_1, ..., X_m) with T constrained to $[T_{min}(f) - \delta, T_{max}(f) + \delta]$ at frequency f. As a result, X_m becomes X_m = [x_{mu}, x_{mb}];

2- Set X_m = [x_{mb}, x_{mb}];

3- *Is_consistent* (T, f, X_1, ..., X_m);

4- If success then print "$x_{m_max}(f) = X_m$" and STOP, else set X_m = [x_{mu}, x_{mb}];

5- Divide X_m into X_L and X_R;

6- Constrain X_m=X_R;

7- *Is_consistent* (T, f, X_1, ..., X_m);

8- If success then push X_L on stack L_u; Go to 16;

9- Constrain X_m = X_L. If the width $w(X_m) > \varepsilon_m$, then go to 5;

10- *Is_consistent* (T, f, X_1, ..., X_m).

11- If success then print "$x_{m_max}(f) = X_m$" (since $w(X_m) =< \varepsilon_m$) and STOP;

12- Else

If L_u is empty, then print "Any deviation of x_m outside its tolerance is observable" and STOP;

13- If L_u is not empty, then constrain X_m = top(L_u) and pop(L_u).

14. *Is_consistent* (T, f, X_1, ..., X_m).

15- If failure then go to 12.

16- If w(X_m) =< ε_m, then print "$x_{m_max}(f) = X_m$" and STOP, else go to 5.

End.

To search for the global minimum $x_{m_min}(f)$, the same algorithm is applied but in a symmetrical manner, that is, we now approach the solution from the left in the domain X_m =[0, x_{mi}] (0 corresponds to the negative deviation of 100% from the nominal parameter value).

4.5 Experimental Results

4.5.1 Example 1

To illustrate our test generation method on a realistic example we analyze the biquadratic filter shown in Figure 4.2 The performances of interest are the magnitudes V_3 and V_5 of the transfer functions at nodes 3 and 5 given by the following equations:

$$V_3 = \frac{1}{R_6 C_2} \frac{\omega}{\sqrt{\left(\frac{R_8}{R_3 R_5 R_7 C_2 C_4} - \omega^2\right)^2 + \frac{\omega^2}{R_1^2 C_2^2}}} \tag{7}$$

$$V_5 = \frac{1}{R_3 R_6 C_2 C_4} \frac{1}{\sqrt{\left(\frac{R_8}{R_3 R_5 R_7 C_2 C_4} - \omega^2\right)^2 + \frac{\omega^2}{R_1^2 C_2^2}}} \tag{8}$$

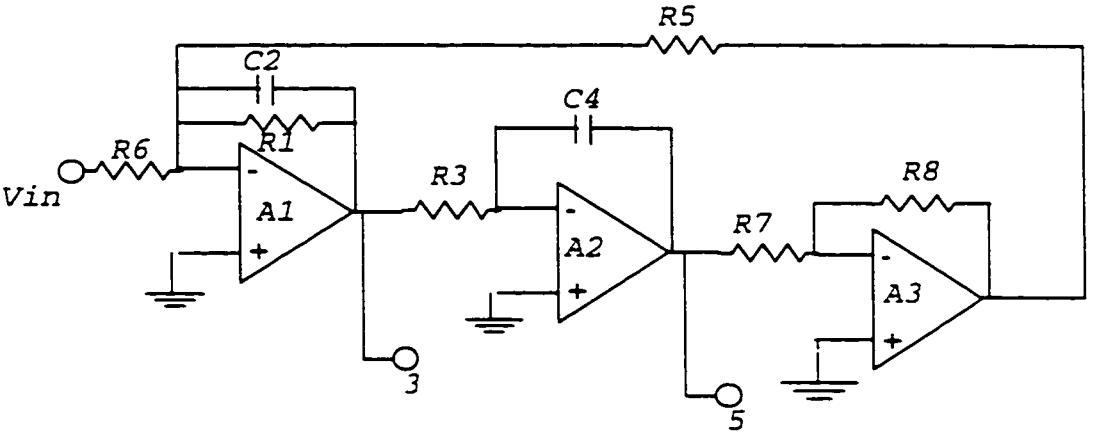


Figure 4.2 A biquadratic filter

The frequency interval of interest is [1Hz, 20Khz], discretized into a set of 35 frequencies. First, the optima $V_{5_max}(f)$, $V_{5_min}(f)$, $V_{3_max}(f)$, and $V_{3_min}(f)$ are computed. These results were compared to those obtained by using the Sequential Quadratic Programming (SQP) standard optimization method from MATLAB [26] as reported in [12], and they match to within 0.2%.

The smallest and largest observable deviations obtained by our method are shown in Table 4.4. Those computed by the SQP optimization are also shown in Table 4.4. The third and the fourth (the sixth and the seventh for SQP) columns give the smallest observable positive deviation (SOPD) and the largest observable negative deviation(LOND) observed at V5 and V3, respectively, and the best test frequency Fr in Hertz determined by our method. From Table 4.4 we can observe that the deviations of the circuit parameters computed by the CLP-based method and by the SQP optimization all match to within 5.4%, except for the LOND of C2 observed at V5 where the value calculated using SQP is in fact a local minimum and thus incorrect.

TABLE 4.4 Results for biquadratic filter

para-meter	nominal value	CLP-based method						SQP optimization			
		magnitude V5			magnitude V3			magnitude V5		magnitude V3	
		SOPD* LOND [%]	Fr [Hz]	CPU time [s]	SOPD LOND [%]	Fr [Hz]	CPU time [s]	SOPD LOND [%]	Fr [Hz]	SOPD LOND [%]	Fr [Hz]
R1	10k	49.07 -31.7	9.55k 9.55k	978.17	27.75 -19.19	10k 10k	49.18	43 -31.6	10k 9.6k	24.9 -23.3	10.300k 10.380k
C2	1.59nF	66.16 -99.9	20 k 1	234.46	35.09 -34	20k 20k	37.65	66.2 -59	20k 9.3k	34.93 -33.05	20 k 20k
R3	10k	45.98 -34.19	9.55k 20k	1420.29	82.60 -45.25	2k 4k	34.58	45.7 -31.2	9.6 k 11.52k	92.4 -45.2	4.020k 4k
C4	1.59nF	45.98 -33.85	9.55k 14k	1183.2	82.60 -45.25	2k 4k	72.92	45.3 -30.3	9.6k 11.04k	92.1 -45	4.1k 4k
R5	10k	43.23 -30.65	1 1	614	82.60 -45.25	2k 4k	34.47	42.9 -30.5	1 1	92.1 -45.2	4k 4k
R6	10k	43.67 -30.30	1 60	702.54	23.43 -18.97	10k 10k	41.37	40.4 -29.1	8.14k 1	27.4 -17.42	11.680k 8.640k
R7	10k	43.06 -30.56	40 200	598.45	82.60 -45.25	2k 4k	34.66	42.9 -30.5	1 1	92.1 -45.2	4k 4k
R8	10k	43.67 -30.30	1 80	582.26	82.18 -45.37	4k 2k	42.67	43.5 -30.2	1 1	82 -48	4k 4k

*SOPD: Smallest Observable Positive Deviation
LOND: Largest Observable Negative Deviation

Note that the quality of results obtained by SQP greatly depends on the starting point, termination criteria, number of iterations, scaling,... which must be selected and introduced by the user by trial and error. Furthermore, these methods use local information (point derivatives) which are not always reliable. Consequently, such standard optimization methods cannot guarantee that the global optimum is found. Furthermore, with SQP (implemented in MATLAB) it took more than a week of hard work to obtain the results. Our method on the other hand is simple, fast, automated, and allows us to easily include other constraints such as correlation between parameters.

4.5.2 Example 2

We analyse here a notch filter due to Bainter [27], as shown in Figure 4.3. The performances of interest are the magnitude and the phase of the transfer function given by the following equation:

$$A(s) = \frac{s^2 + \dfrac{R_2}{R_1 R_4 R_7 C_3 C_5}}{s^2 + \dfrac{R_7 + R_8}{R_7 R_8 C_3} s + \dfrac{1}{R_6 R_7 C_3 C_5}} \tag{9}$$

The nominal design has the notch frequency at $f_0 = 275.6$Hz. Figure 4.4 shows the nominal response and the worst-case values of |A(s)|. The total CPU time for computing the worst-case magnitude and phase at 22 frequencies in the interval [1, 10KHz] is about 300s and 130s, respectively.

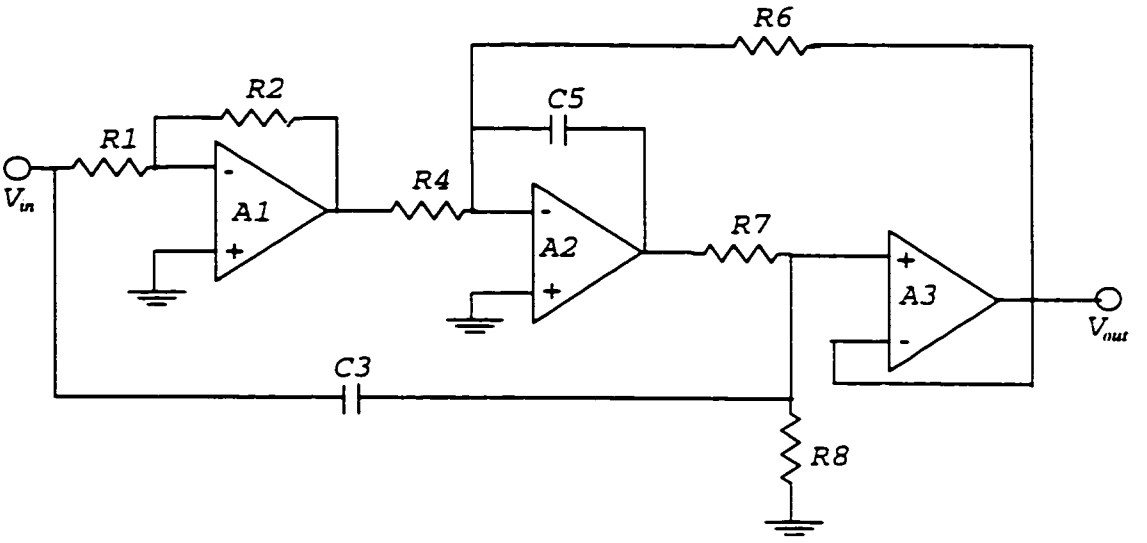


Figure 4.3 A notch filter

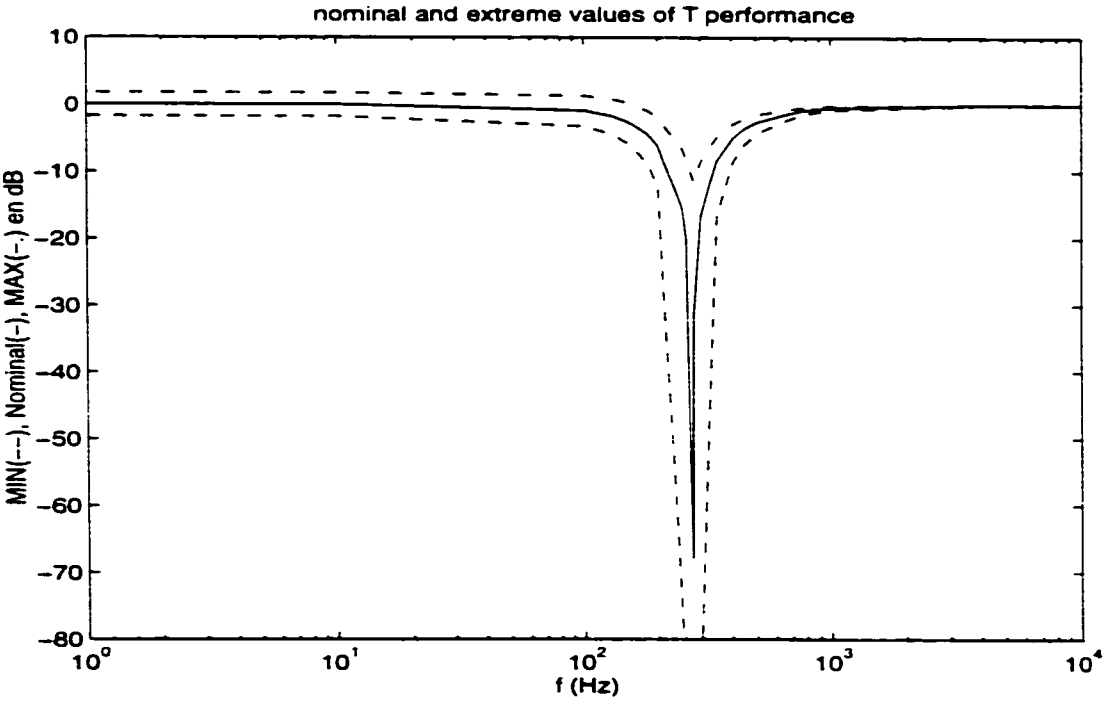


Figure 4.4 Nominal and worst-case magnitude of notch filter

The smallest (largest) observable deviations of the circuit parameters obtained by our method are given in Table 4.5.

TABLE 4.5 Results for notch filter

parame-ter	nominal value	CLP-based method							
		magnitude |A|			phase FA			Final test frequency	Best
		SOPD* LOND [%]	Fr [Hz]	CPU time [s]	SOPD LOND [%]	Fr [Hz]	CPU time [s]	Fr [Hz]	SOPD LOND [%]
R1	5774 ohms	44.2 -30.58	1 1	614	not observable			1 (|A|) 1 (|A|)	44.2 -30.58
R2	5774 ohms	43.42 -30.8	1 1	644	not observable			1 (|A|) 1 (|A|)	43.42 -30.8
C3	0.1 μ F	58 -33.73	350 503	1136	24.07 -19.37	755 1001	488	755 (FA) 1001(FA)	24.07 -19.37
R4	3337 ohms	44 -30	1 1	596	not observable			1 (|A|) 1 (|A|)	44 -30
C5	0.1 μ F	97.47 -44	150 275.6	568	36.5 -26.89	150 180	232	150 (FA) 180 (FA)	36.5 -26.89
R6	3337 ohms	43 -31	1 1	266	36.5 -26.88	150 180	242	150 (FA) 180 (FA)	36.5 -26.88
R7	9989 ohms	77 -76	280 4001	1826	41.81 -30.5	250 450	2157	250 (FA) 450 (FA)	41.81 -30.5
R8	9989 ohms	$> 10^{4}$ -54	1 1001	559	95.77 -36.67	1001 1001	1786	1001(FA) 1001(FA)	95.77 -36.67

*SOPD: Smallest Observable Positive Deviation
LOND: Largest Observable Negative Deviation

Note that all the results were computed with $\varepsilon = 10^{-4}$ and by assuming parameter tolerances and the resolution of the test equipment to be ±5% . and 0.01V for magnitude and 1 degree for phase measurements, respectively.

4.5.3 Test quality

The quality of a generated test depends on its ability to maximize the detectability of a fault. I.e., it depends on how small is the smallest observable deviation of a

parameter. This value depends on the observed performance, the accuracy of the method which is used to find these optima, the frequency range, the selected test frequency, the tolerance intervals of the parameters, the resolution of the test equipment, etc.

4.5.3.1 Effects of tester resolution

The resolution of the test equipment used to measure a performance may affect the quality of a test by reducing the detectability of a fault. To illustrate the impact of tester resolution, consider the biquad filter with a quality factor Q = 20. The corresponding computed test data are summarized in Table 4.6. We can observe from this table that the change in detectability (SOPD, LOND) introduced by the tester resolution varies from 1% to 186%. The sets of test frequencies S and S_{TR} extracted from Table 4.6 that correspond to the cases with and without taking into account tester resolution, respectively, are:

$$S = \{1, 401, 601, 803, 1001, 10001, 20K\}_{V5} \cup \{20K\}_{V3} \quad \text{[Hz]}$$

$$S_{TR} = \{1, 103, 4K, 9K, 10001, 10.5K\}_{V5} \cup \{10001, 12.5K, 20K\}_{V3} \quad \text{[Hz]}$$

Recall that S and S_{TR} detect any fault deviation of parameters outside their]LOND, SOPD[range. We can see from Table 4.6 that S and S_{TR} guarantee a fault coverage of 100% for parameter faults outside the ranges]-86.9, 47.64[% and]-87.76, 88.39[% respectively. As a result, owing to the tester resolution of 0.01V, parameter faults in the range]-87.76, -86.9]% $\cup$ [47.64, 88.39[% are not guaranteed to be

detected. Consequently, In such cases a testeur with an appropriate resolution is required.

TABLE 4.6 Tester resolution impact on parameter detectability

para meter	nominal value	magnitude without taking into account tester resolution				magnitude with tester resolution of 0.01V			
		SOPD LOND [%]		Test frequency	Best SOPD LOND	SOPD LOND [%]		Test frequency	Best SOPD LOND
		V5	V3	Fr [Hz]	[%]	V5	V3	Fr [Hz]	[%]
R1	200k	*	*	$\forall Fr$	*	*	*	$\forall Fr$	*
		-86.9	-86.9	10001(V5)	-86.9	-87.76	-87.85	10001(V5)	-87.76
C2	1.59nF	47.67	26.72	20K(V3)	26.72	67.77	53.66	12.5K(V3)	53.66
		-34	-23.7	20K(V3)	-23.7	-48.02	-34.64	20K(V3)	-34.64
R3	10k	47.64	73.19	20K(V5)	47.64	67.35	105.38	10.5K(V5)	67.35
		-33.98	-42.4	20K(V5)	-33.98	-55.1	-43.65	10001(V3)	-43.65
C4	1.59nF	47.63	73.18	20K(V5)	47.63	67.35	105.38	10.5K(V5)	67.35
		-33.98	-42.4	20K(V5)	-33.98	-47.94	-43.65	10001(V3)	-43.65
R5	10k	42.27	73.19	1001(V5)	42.27	62.76	105.38	4K(V5)	62.76
		-29.87	-42.4	803(V5)	-29.87	-40.23	-43.65	9K(V5)	-40.23
R6	200k	42.27	42.99	1(V5)	42.27	88.39	122.94	103(V5)	88.39
		-29.9	-30.24	1(V5)	-29.9	-39.76	-43.66	1(V5)	-39.76
R7	10k	42.22	73.19	601(V5)	42.22	62.76	105.38	4K(V5)	62.76
		-29.81	-42.4	803(V5)	-29.81	-40.23	-43.65	9K(V5)	-40.23
R8	10k	42.12	73.19	803(V5)	42.12	66.88	77.02	9K(V5)	66.88
		-29.85	-42.4	401(V5)	-29.85	-38.71	-51.43	4K(V5)	-38.71
* undetectable									

4.5.3.2 Effects of parameter tolerances

One of the relevant factors that affect the quality of a test is the width of parameter tolerances. Indeed, the more the manufacturing process is improved and the tolerance intervals are smaller, the more their impact on the quality of a test is minimized. To illustrate this fact, consider again the notch filter. SOPD (LOND) computed with parameter tolerances of $\pm 2\%$ and $\pm 5\%$ are given in the third and the fourth columns

of Table 4.7. The test sets S_2 and S_5 obtained with parameter tolerances of ±2% and ±5% respectively, as extracted from Table 4.7 are as follows:

$$S_2 = \{1\}_{|A|} \cup \{160, 180, 260, 300, 503, 755, 1001\}_{FA} \text{ [Hz]}$$

$$S_5 = \{1\}_{|A|} \cup \{150, 180, 250, 450, 755, 1001\}_{FA} \text{ [Hz]}$$

We can observe from Table 4.7 that S_2 and S_5 guarantee a fault coverage of 100% for faults outside the ranges]-22.53, 36.31[% and]-36.67, 95.77[%, respectively. As a result, owing to the change of parameter tolerances from ±2% to ±5%, any fault in the range]-36.67, -22.53]% ∪ [36.31, 95.77[%is not guaranteed to be detected. Consequently, the quality of a test can be improved by using manufacturing process with smaller parameter tolerances.

TABLE 4.7 Parameter tolerances impact on parameter detectability

para meter	nominal value	magnitude |A| SOPD LOND [%]		phase FA SOPD LOND [%]		(Fr [Hz] (|A| or FA), SOPD or LOND [%])	(Fr [Hz] (|A| or FA), SOPD or LOND [%])
		tolerance = ±2%	±5%	tolerance = ±2%	±5%	tolerance = ±2%	tolerance = ±5%
R1	5774Ω	16.5 -14	44.2 -30.58	not observable		(1(|A|), 16.5) (1(|A|), -14)	(1 (|A|), 44.2) (1 (|A|), -30.58)
R2	5774Ω	16.2 -14.1	43.42 -30.8	not observable		(1(|A|), 16.2) (1(|A|), -14.1)	(1 (|A|), 43.42) (1 (|A|), -30.8)
C3	0.1μF	20.58 -16.39	58 -33.73	11.4 -10.25	24.07 -19.37	(503(FA), 11.4) (755(FA), -10.25)	(755 (FA), 24.07) (1001(FA), -19.37)
R4	3337Ω	16.5 -14	44 -30	not observable		(1(|A|), 16.5) (1(|A|), -14)	(1 (|A|), 44) (1 (|A|), -30)
C5	0.1μF	28.5 -20.4	97.47 -44	11.48 -12.9	36.5 -26.89	(160(FA), 11.48) (180(FA), -12.9)	(150 (FA), 36.5) (180 (FA), -26.89)
R6	3337Ω	16.2 -14.2	43 -31	14.8 -12.9	36.5 -26.88	(160(FA), 14.8) (180(FA), -12.9)	(150 (FA), 36.5) (180 (FA), -26.88)
R7	9989Ω	25.8 -71.2	77 -76	17.48 -14.8	41.81 -30.5	(260(FA), 17.48) (300(FA), -14.8)	(250 (FA), 41.81) (450 (FA), -30.5)
R8	9989Ω	130.5 -35	$> 10^4$ -54	36.31 -22.53	95.77 -36.67	(755(FA), 36.31) (1001(FA), -22.53)	(1001 (FA), 95.77) (1001 (FA), -36.67)

4.6 Acceleration techniques

The algorithms we have proposed so far are exhaustive, that is, they are based on discarding those regions of the search space where the constraint system is inconsistent, that is, where there is no solution. However, from Tables 4.4 and 4.5 we can see that the average CPU times per optimization (i.e., at one frequency) related to the biquad and notch circuits are about 20s and 6s, respectively, which is quite reasonable. Nevertheless, we propose here two techniques that can be included in Algorithms 1 and 2 to further speed-up the computation.

4.6.1 Monotonicity test

Monotonicity test is an efficient known technique [20] for determining wether a function T is strictly monotone on an interval X_i, i=1, ...m. If this is the case, then the interior of X_i cannot contain the global optimum. The interval X_i can be reduced to one of its bounds, which in turn reduces the search space.

4.6.1.1 Computation of the worst-case circuit response

In this case, the monotonicity test can be added to Algorithm 1 after Step 4 and applied as follows. If the derivative $D_i = \frac{\partial T_i}{\partial X_i}(X_1, X_2, \ldots, X_i, \ldots, X_m) = T_i'(X) = [d_{il}, d_{iu}]$, i=1.....m, is positive ($d_{il} > 0$) or negative ($d_{iu} < 0$), the performance T is increasing or decreasing over the interval X_i. Then, for computing the lower bound of T one of the bounds of X_i is used, while the other bound is used for computing the upper bound of T. If all intervals degenerate in this way, then the solution is directly obtained by eval-

uating the relation T == T(f,X_1,..., X_m). Nevertheless, even if only some of the intervals are degenerated, it still may speed up the computation since the search space is reduced.

To illustrate the effectiveness of the monotonicity test in speeding-up the computation, let's consider the computation of the worst-case response of the biquad filter. The results computed by Algorithm 1 and by its modified version including the monotonicity test are shown in Table 4.8. We can observe from this table that the CPU-time is reduced by a factor of 2 to 4 when the monotonicity test is used.

TABLE 4.8 Effect of monotonicity test

Biquad filter (Q=1)	Total CPU time [s] (for all 35 frequencies)	
	without monotonicity test	with monotonicity test
Worst-case magnitude V5	406	96
Worst-case magnitude V3	86	39

4.6.1.2 Computation of observable parameter values

We address now the question of how to apply the monotonicity test for speeding up the computation of $x_{m_max}(f)$ (resp. $x_{m_min}(f)$).

Consider the problem of seeking $x_{m_max}(f)$ (equation 3 Section 2). Recall that the initial domain $X_m = [x_{mu}, \bar{x}_{mu}] = [x_{mu}, +\infty]$ of the parameter x_m is reduced to $X_m = [x_{mu}, x_{mb}]$ by the evaluation of the relation T == T(f,X_1,..... X_m) (Step 1 of Algorithm 2). Considering that an open resistance is usually modeled as 100 Mohms and a shorted capacitance is modeled as 1F, we can compute the derivative $D_m = T_m'(X)$ over $X_m = [x_{mu}, 10^8]$ for a resistance and $X_m = [x_{mu}, 1]$ for a capacitance.

If, for a given frequency, the derivative $D_m = T_m'(X)$ is positive or negative, the function is increasing or decreasing over the entire interval X_m. Then, if $D_m > 0$ (resp. $D_m < 0$), maximizing x_m under the worst effects of the other parameters may be viewed as minimizing (resp. maximizing) T under the same constraints. This stems from the fact that x_m on one hand and the other parameters on the other hand should have the opposite effects on T. Consequently, if $D_i = T_i'(X)$, i=1, ...(m-1) is positive or negative, then X_i can be degenerated to one of its bounds. If all the intervals X_i, i=1, ...,(m-1), so degenerate, then $x_{m_max}(f)$ is directly obtained by evaluating the relation T == T(f,X_1,...., X_m), and the Algorithm 2 can be stopped.

To illustrate the impact of the monotonicity test on speeding up the computation of $x_{m_max}(f)$ and $x_{m_min}(f)$ consider again the biquad circuit. We modified Algorithm 2 to include the monotonicity test after Step 4. Table 4.9 shows the results computed by the original Algorithm 2 (from Table 4.4) and those computed by using the modified version. It can be seen from Table 4.9 that the use of monotonicity test in combination of the improved mean-value form (a particular form of interval extension which is given below) allows to reduce the CPU time by up to 41.5%.

4.6.2 Mean-value form

An interval function may be calculated by the mean-value form as follows [27] [28], assuming continuous first derivatives. Let $f(\mathbf{x}) = f(x_1,....x_m)$ be a rational function, F(X) =F(X_1, ..., X_m) be its natural interval extension, and $F_{MV}(X)$ be its mean-value form. Let $F_i'(X)$ be the interval extension of $\frac{\partial}{\partial x_i} f(x) = f_i'(x)$, i=1....,m, and **y** the center (midpoint) vector of X. Then,

$$F_{MV}(X, x) = f(y) + \sum_{i=1}^{m} F_i'(X)(X_i - y_i) \tag{10}$$

For smaller intervals the mean-value form $F_{MV}(X)$ provides sharper bounds than the natural extension F(X) [28]. On the other hand, this is not true for larger boxes. In practice, the intersection $F(X) \cap F_{MV}(X)$ produces the best results.

An even better mean-value interval extension of f(**x**) called the improved monotonicity test (MT) form was proposed in [25]:

$$F_{MT}(X, x^L) = f(x^L) + \sum_{i=1}^{m} F_i'(X)(X_i - x_i^L) \tag{11}$$

and

$$F_{MT}(X, x^U) = f(x^U) + \sum_{i=1}^{m} F_i'(X)(X_i - x_i^U) \tag{12}$$

where $F_i'(X) = [a_i, b_i]$, $X_i = [x_{il}, x_{iu}]$, $x^L = (x_1^L, \ldots, x_m^L) \in X$ such that $x_i^L = x_{il}$ if $a_i \geq 0$, $x_i^L = x_{iu}$ if $b_i \leq 0$, or $x_i^L = \frac{(b_i x_{il} - a_i x_{iu})}{b_i - a_i}$ if $i \in Z$, with Z being the set of integers such that $F_i'(X)$ contains zero. Also, $x^U = (x_1^U, \ldots, x_m^U) \in X$ such that $x_i^U = x_{iu}$ if $a_i \geq 0$, $x_i^U = x_{il}$ if $b_i \leq 0$, or $x_i^U = \frac{(b_i x_{iu} - a_i x_{il})}{b_i - a_i}$ if $i \in Z$.

The improved monotonicity test form provides even sharper bounds and may thus further speed up the computation. In particular, it is not expensive when combined with the monotonicity test since it uses the same derivatives of T.

The results in Table 4.9 were obtained using Algorithm 2 modified to compute $T = T(X) \cap T_{MT}(X)$ in conjunction with the monotonicity test.

TABLE 4.9 Speeding-up the computation of SOPD and LOND values of biquad parameters

parameter	nominal value	Magnitude response V5	Total CPU-time [s] (for all 35 frequencies)	
		SOPD LOND (%)	without monotonicity test	with monotonicity test
R1	10k	49.07 -31.7	978.17	690
C2	1.59nF	66.16 -99.9	234.46	137
R3	10k	45.98 -34.19	1420.29	885
C4	1.59nF	45.98 -33.85	1183.2	827
R5	10k	43.23 -30.65	614	434.5
R6	10k	43.67 -30.30	702.54	530
R7	10k	43.06 -30.56	598.45	433
R8	10k	43.67 -30.30	582.26	376

4.6.3 Reducing search space of parameters

Maximizing the quality of a test means maximizing the detectability of a fault. That is, the smaller the smallest observable deviation of a parameter is, the better is its detectability, and the quality of the corresponding test is thus also better. Assume now, we can establish a detectability threshold that would allow us to decide when to apply design for testability (DFT) techniques. More precisely, if the smallest (resp. largest) observable positive (resp. negative) deviation of a parameter x_m is larger (resp. smaller) than a detectability threshold dt_m^+ (resp. dt_m^-), then DFT should be used

to enhance its detectability. Consequently, we can use the threshold to reduce the search space of parameter x_m to within $DT_m = [dt_m^-, dt_m^+]$. However, once a solution is found inside DT_m we should verify that the constraint system is inconsistent outside this interval. This guarantees that the found solution is global. This reduction of the parameter search space may further speed up the computation.

4.7 Conclusions

In this paper, we proposed new CLP-based algorithms for solving the optimization problems that are used in the test generation and worst-case tolerance analysis for linear analog circuits. The algorithms find tight and guaranteed bounds on the global optima of a performance or a parameter. This quality as achieved by our algorithms stems from the inclusion properties of intervals, the correctness of narrowing process, and the outward rounding on which are based the relational interval arithmetic and CLP. The effectiveness of our algorithms was illustrated on a number of nonlinear functions known to be difficult. Then, our test generation method was applied to the biquad and notch benchmark filter circuits. The results of the former were compared with those obtained by using the SQP standard optimization method. The results match in all cases except one for which SQP fell into a local minimum and thus produced incorrect results. Contrary to standard optimization methods which are non-automatic, the proposed CLP-based algorithms are automatic and fast.

We also showed how the resolution of test equipment and the quality of the manufacturing process impact on the quality of a test. It was seen that with better resolution and smaller parameter tolerances, the detectability of circuit parameter variations

increases and the test quality is thus better. Finally, we demonstrated that the use of the monotonicity test and the improved mean-value form may speed up the computation.

Our further research is now directed toward generalizing our method to non-linear circuits.

References

[1] Semiconductor Industry Technology Workshop Conclusions, Semiconductor Industry Association, 1993.

[2] European Design and Test Conference, panel discussion on DFT, Mars 1996.

[3] L. Milor and V. Visvanathan. "Detection of catastrophic faults in analog integrated circuits," IEEE Trans. Computer-Aided Design, vol.8, pp.114-130, Feb. 1989.

[4] L. Milor and A. Sangiovanni-Vincentelli, "Optimal Test Set Design for Analog Circuits", Proc. ICCAD, pp. 294-297, 1990.

[5] L. Milor and A. Sangiovanni-Vincentelli, "Minimizing Production Test Time to Detect Faults in Analog Circuits", IEEE Trans. Computer-Aided Design, Vol.13, No.6, pp.796-813, June 1994.

[6] Shen-Jen Tsai, "Test Vector Generation for Linear Analog Devices", Proc. IEEE International Test Conference, pp. 592-597, 1990.

[7] N. Ben Hamida and B. Kaminska, "Analog Circuit Testing Based on Sensitivity Computation and New Circuit Modeling", Proc. IEEE International Test Conference, pp. 652-661, 1993.

[8] N. Ben Hamida, K. Saab, D. Marche, B. Kaminska and G. Quesnel, "LIMSoft: Automated Tool for Design and Test Integration of Analog Circuits", IEEE International Test Conference, pp. 571-580, 1996.

[9] N. Nagi, A. Chatterjee, A. Balivada and J. Abraham, "Fault-based Automatic Test Generator for Linear Analog Circuits", Proc. of ICCAD, pp. 88-91, 1993.

[10] M. Slamani and B. Kaminska, "An Integrated Approach for Analog Circuit Testing with Minimum Number of Detected Parameters", Proc. IEEE International Test Conference, pp. 631-649, 1994.

[11] G. Devarayanadurg and M. Soma, "Analytical Fault Modeling and Static Test Generation for Analog ICs", Proc. of ICCAD, pp. 44-47, 1994.

[12] A. Abderrahman, E. Cerny and B. Kaminska, "Optimization-based Multifrequency Test Generation for Analog Circuits", Journal of Electronic Testing (JETTA), vol. 9, no 1/2, pp. 59-73, August/October 1996.

[13] J. Jaffar and M. Maher,"Constraint Logic Programming: A Survey", Journal of Logic Programming, 19/20, 503-581, 1994.

[14] R.E. Moore,"*Interval Analysis*". Prentice Hall, 1966.

[15] R.E. Moore,"*Methods and Applications of Interval Analysis*", Philadelphia, SIAM, 1979.

[16] J.G. Cleary,"Logical Arithmetic", Future Computing Systems, Vol. 2,No 2, pp.125-149, 1987.

[17] W.J. Older, "Introduction to Constraint Logic Programming" Tutorial given at Inter-University Center in Computer Architecture and VLSI, Université de Montréal, 1995.

[18] W. Older and A. Vellino, "Constraint Arithmetic on Real Intervals", Constraint Logic Programming: Selected Research, 1993.

[19] A.K. Mackworth, "Consistency in Network of Relations", Artificial Intelligence 8, pp.99-118, 1977.

[20] S. Skelboe, "True Worst-Case Analysis of linear electrical circuits by interval arithmetic", IEEE Transactions on Circuits and Systems, Vol. CAS-26, pp. 874-879, october 1979.

[21] CLP(BNR) Prolog User Guide and Reference Manual, BNR 1988.

[22] E. Hassen, "Global Optimization Using Interval Analysis: The Multi_Dimensional Case", Numerische MathematiK, Vol. 34, pp. 247-270, 1980.

[23] N. S. Asaithambi, Shen Zude, R.E. Moore, "On Computing the Range of Values", Computing, Vol. 28, pp. 225-237, 1982.

[24] R.E. Moore, H. Ratschek, "Inclusion Functions and Global Optimization II", Mathematical Programming, 1987.

[25] L. V. Kolev et al., "Interval Mathematics Algorithms for Tolerance Analysis", IEEE Transaction on Circuits and Systems, Vol. 35, No 8, pp. 967-975, August 1988.

[26] Optimization Toolbox User's Guide, December 1992, MathWorks, Inc.

[27] J.R. Bainter, "Active Filter Has Stable Notch, and Response can be Regulated", Electronics, pp.115-117, Oct. 1975.

[28] S. Skelboe, "Computation of Rational Interval functions", BIT 14, pp. 87-95, 1974.

Chapitre 5
Analyse et amélioration de la détectabilité des défauts

5.1 Introduction

Dans ce chapitre nous nous proposons d'appliquer la méthode de génération de tests, décrite dans les chapitres 3 et 4, à l'analyse de testabilité des circuits analogiques linéaires. Grâce à cette analyse, nous pouvons décider s'il est nécessaire de recourir à la technique de design orienté testabilité (DFT) pour améliorer la testabilité du circuit considéré.

L'analyse de testabilité des circuits joue un rôle important dans la pratique de la technique DFT. Dans l'univers numérique, l'objectif fondamental de cette technique consiste à augmenter la contrôlabilité et l'observabilité des noeuds en modifiant le circuit considéré. Pour atteindre cet objectif, il faut disposer d'une mesure de testabilité qui nous permet de déterminer les noeuds ayant une faible testabilité, et de modifier le circuit en conséquence en vue de les rendre facilement testables. Or, cette modification peut impliquer une augmentation du nombre de broches d'entrées-sorties et un ajout de circuiterie qui se traduit par une augmentation de la surface et de la complexité du circuit. Il va de soi, qu'une méthode imprécise de mesure de testabilité peut

conduire à la conclusion qu'un noeud a une faible testabilité alors qu'en réalité ce n'est pas le cas, et vice-versa. Par conséquent, toute mesure de testabilité inadéquate peut impliquer une augmentation inutile du nombre de broches du circuit et de sa surface, se traduisant entre autres, par une augmentation de la puissance consommée et une diminution du rendement du procédé de fabrication. Il en découle, que pour tirer le maximum de profit de la technique DFT, il est nécessaire de disposer d'une méthode suffisamment précise de mesure de testabilité.

Dans ce contexte, nous procéderons dans ce chapitre, à l'analyse de détectabilité des paramètres dans les circuits analogiques grâce à l'application de la méthode de génération de tests, proposée dans cette thèse. Mais tout d'abord, introduisons brièvement certaines définitions qui seront utiles pour la suite de cet exposé.

5.2 Définitions

Nous avons vu dans les chapitres précédents qu'étant donné un noeud et une performance d'observation, chaque paramètre du circuit sous-test est caractérisé par deux quantités SOPD (la plus petite déviation positive observable) et LOND (la plus grande déviation négative observable). Ces quantités permettent de mesurer l'importance de l'étendue de la plage où les défauts sont observables. En effet, rappelons que toute déviation du paramètre en dehors de la plage]LOND, SOPD[est garantie d'être détectée par l'ensemble de tests générés. Dans ce qui suit nous allons proposer une

mesure de détectabilité des déviations d'un paramètre. Ensuite, nous définissons les conditions d'indétectabilité d'un paramètre défectueux.

Définition 1: degré de détectabilité

Le degré d_i de détectabilité des déviations paramétriques d'un paramètre x_i caractérisé par les quantités $(SOPD)_i$ et $(LOND)_i$, et de tolérance $\pm p_i$ [%], est égal à:

$$d_i = 100e^{-\left(\frac{1+v_i}{1+p_i}-1\right)} \quad [\%] \tag{4}$$

avec $v_i = (SOPD)_i$ pour les déviations positives et $v_i = |(LOND)_i|$ pour les déviations négatives. Nous remarquons que lorsque $v_i = (SOPD)_i = |(LOND)_i|$ tend vers p_i, le degré de détectabilité du paramètre x_i tend vers 100%. Par contre, quand v_i tend vers l'infini, le degré de détectabilité de x_i tend vers zéro.

Dans certains circuits, il arrive que les déviations d'un paramètre soient indétectables quelques soient leurs valeurs. Selon les auteurs de [34] un défaut est inobservable (indétectable) à un noeud, si pour toutes les fréquences, le signal observé en ce noeud a une sensibilité nulle relativement à ce défaut. Cependant, vu que la sensibilité est une approximation du premier ordre, il n'est pas garanti qu'un défaut inobservable soit déterminé comme tel.

Nous introduisons ici une nouvelle définition de l'indétectabilité d'un paramètre:

Définition 2: **Indétectabilité absolue**

Étant donné une performance T à observer, un paramètre défectueux est absolument indétectable, si quelque soit la valeur de sa déviation et quelque que soient les valeurs tolérées prises par les autres paramètres, T reste dans sa plage d'acceptance quelque soit le test appliqué.

Définition 3: **Indétectabilité simple**

Étant donné un noeud de test et une performance T à observer, un paramètre défectueux est indétectable, s'il existe au moins une combinaison des valeurs tolérées prises par les autres paramètres tel que T reste dans sa plage d'acceptance quelque soit la valeur du défaut et quelque soit le test appliqué.

L'indétectabilité simple est plus probable d'arriver que l'indétectabilité absolue, car les contraintes associées sont moins fortes. En effet, il suffit qu'une seule combinaison des valeurs tolérées des autres paramètres puisse masquer les effets du paramètre défectueux sur T quelque soient la valeur du défaut et le test appliqué, pour que ce défaut soit déclaré indétectable.

Dans le contexte de l'Algorithme 2, décrit dans le chapitre 4, l'indétectabilité simple se traduit par le fait que le système des contraintes reste consistant quelque soit la valeur de la déviation du paramètre défectueux et quelque soit le test (fréquence)

appliqué. Autrement dit, il existe une combinaison des valeurs des autres paramètres pour laquelle les contraintes du système sont satisfaites.

Dans la suite de ce chapitre, seulement l'indétectabilité simple sera considérée, et elle sera désignée par "Indétectabilité" tout court.

Maintenant que nous disposons d'une mesure de détectabilité (définition 1 ci-dessus), nous pouvons procéder à l'application de la méthode de génération de tests proposée, à l'analyse de détectabilité des circuits analogiques en l'illustrant sur certains exemples.

5.3 Analyse de détectabilité du filtre biquadratique

Dans cette section, nous nous proposons d'analyser la détectabilité de chacun des paramètres du filtre biquadratique de la figure 5.13.

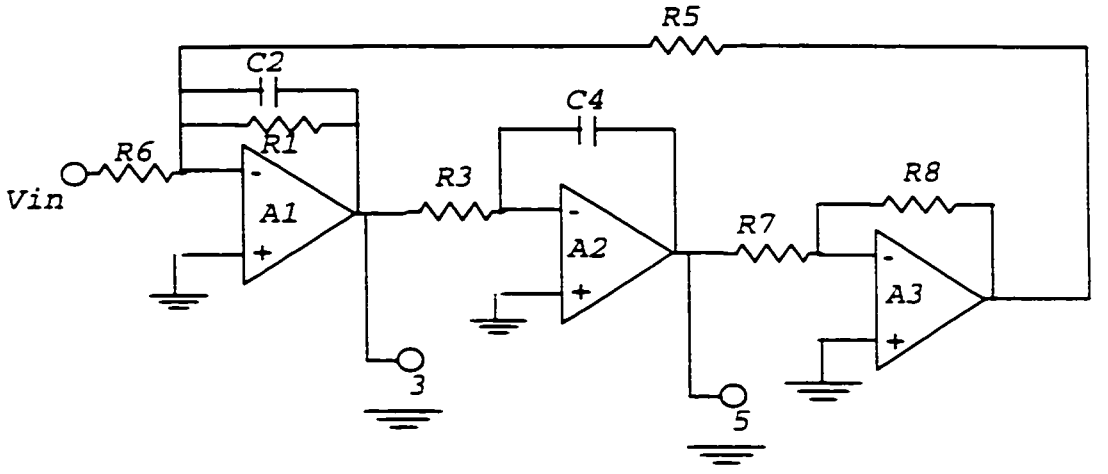


Figure 5.13: Filtre biquadratique

5.3.1 Observation des gains V3 et V5

Dans le cas où les performances à observer sont les gains V3 et V5 (noeuds 3 et 5), l'application de la méthode de génération de tests produit les résultats indiqués dans le tableau 5.1. Ces résultats ont été obtenus pour des tolérances de ±5% et une résolution de l'appareil de mesure de 0.01V.

TABLEAU 5.1 Résultats obtenus en observant les gains V3 et V5

paramètre	valeur nominale	V5		V3	
		SOPD LOND [%]	détectabilité d [%]	SOPD LOND [%]	détectabilité d [%]
R1	200k	indétectable*		indétectable*	
		-87.76	45.46	-87.85	45.43
C2	1.59nF	67.77	55	53.66	62.91
		-48.02	66.38	-34.64	75.41
R3	10k	67.35	55.22	105.38	38.44
		-55.1	62.05	-43.65	69.2
C4	1.59nF	67.35	55.22	105.38	38.44
		-47.94	66.43	-43.65	69.2
R5	10k	62.76	57.69	105.38	38.44
		-40.23	71.49	-43.65	69.2
R6	200k	88.39	45.19	122.94	32.52
		-39.76	71.82	-43.66	69.2
R7	10k	62.76	57.69	105.38	38.44
		-40.23	71.49	-43.65	69.2
R8	10k	66.88	55.47	77.02	50.36
		-38.71	72.54	-51.43	64.26
*quelque soit la fréquence					

Le résultat le plus remarquable de ce tableau est que toute déviation positive de R1 est indétectable (n'est observable ni à V3 ni à V5) quelque soit la fréquence du signal d'excitation appliqué.

5.3.2 Observation des phases aux noeuds 3 et 5

Voyons maintenant s'il est possible de détecter les déviations de R1 en observant les phases des signaux aux noeuds 3 et 5. D'après le tableau 5.2, nous constatons que les déviations positives de R1, qui sont garanties d'être détectables en observant la phase des signaux aux noeuds 3 et 5, sont celles qui sont supérieures à 179.96% et 252.92% respectivement. Par conséquent, les déviations positives de R1, qui sont indétectables en observant V3 et V5, passent à une détectabilité de 18.89% et de 9.43% en observant la phase aux noeuds 3 et 5 respectivement. Bien qu'il s'agit d'une amélioration notable, toutefois cela reste insuffisant.

TABLEAU 5.2 : Résultats obtenus en observant les phases aux noeuds 3 et 5

		Phase au noeud 3				Phase au noeud 5			
para-mètre	**valeur nominale**	**SOPD LOND [%]**	**d** [%]**	**Fr [Hz]**	**temps CPU* [s]**	**SOPD LOND [%]**	**d** [%]**	**Fr [Hz]**	**temps CPU* [s]**
R1	200 K	179.96 -49.35	18.89 65.54	14K 20K	20	252.92 -50.8	9.43 64.65	14K 20K	363
**détectabilité. *pour l'ensemble des 35 fréquences									

5.3.3 Observation du facteur Q

Dans l'étape suivante nous allons voir s'il est possible d'obtenir davantage une meilleure détectabilité de R1 en observant le facteur de qualité du circuit Q:

$$Q = R_1 C_2 \omega_0 = R_1 C_2 \sqrt{\frac{R_8}{R_3 R_5 R_7 C_2 C_4}} \quad (5)$$

En appliquant la méthode de génération de tests (décrite dans les chapitres 3 et 4) pour une fréquence fixée, nous obtenons alors les valeurs LOND= -29.68% et SOPD = 41.83%, soit une détectabilité de 79.05% et de 70.41% respectivement. Par conséquent, l'observation de Q permet une augmentation importante de la détectabilité des déviations de R1. A titre d'exemple, la détectabilité des déviations positives passe de 18.89%, en observant la phase au noeud 3, à 70.41% en observant Q. Il s'en suit qu'en appliquant la méthode de génération de tests à l'analyse de détectabilité en combinaison avec l'augmentation des points de tests potentiels et l'exploration des différentes performances associées, on peut obtenir une meilleure détectabilité des paramètres. La question qui s'impose à ce niveau est: comment peut-on aboutir à la meilleure détectabilité des paramètres sans l'obligation de procéder à une exploration exhaustive de l'ensemble de performances et de points de test potentiels? Cependant, le sujet de cette question sort de la portée de ce travail et fera l'objet d'un travail futur.

5.3.4 Application de la technique DFT

Bien que l'observation du facteur Q garantit une meilleure détectabilité de R1 comparativement aux autres performances considérées, sa mesure peut coûter relativement chère. D'autre part, on peut être tenté d'obtenir encore une meilleure

détectabilité des paramètres du circuit en appliquant la technique DFT. En effet, dans le circuit du filtre biquadratique de la figure 19, si l'on coupe le circuit de réaction au moyen d'un interrupteur qu'on suppose ici parfait (en série avec R5), alors l'expression du gain au noeud 3 se simplifie et devient:

$$V_3 = \frac{R_1}{R_6} \frac{1}{\sqrt{1 + R_1^2 C_2^2 \omega^2}} \tag{6}$$

L'observation de V3 permet d'analyser la détectabilité des paramètres R1, R6 et C2. Les résultats indiqués dans le tableau 5.3 sont calculés pour les mêmes tolérances et la même résolution qu'auparavant. D'après les résultats de ce tableau obtenus en observant V3, et par comparaison aux résultats obtenus en observant Q, nous constatons que l'application de la technique DFT a permis d'améliorer la détectabilité des déviations de R1 en passant de 79.05% à 90.92% pour les déviations négatives et de 70.41% à 89.11% pour les déviations positives.

TABLEAU 5.3 : Résultats obtenus en observant V3

		V3			
paramètre	**valeur nominale**	**SOPD LOND [%]**	**détectabilité d [%]**	**Fr [Hz]**	**temps CPU [s] (pour 35 fréq.)**
R1	200Kohms	17.1 -14.99	89.11 90.92	1 1	24
C2	1.59 nF	23.08 -18.73	84.18 87.74	2K 2K	182
R6	200Kohms	17.35 -14.82	88.90 91.07	1 1	412

Nous obtenons également une amélioration importante des détectabilités des paramètres C2 et R6 comparativement à celles obtenues en observant les gains aux noeuds 3 et 5 (tableaux 5.1 et 5.3).

Bien entendu, nous pouvons encore pousser l'analyse et observer les phases aux noeuds 3 et 5 ainsi que le gain V5 du filtre modifié (incluant un interrupteur dans le circuit de réaction). On peut s'attendre à une meilleure détectabilité de l'ensemble des paramètres du circuit.

Sur un autre plan, l'interrupteur n'est pas un composant parfait. Les caractéristiques les plus importantes d'un interrupteur sont sa résistance de commutation R_{ON} et sa résistance de blocage R_{OFF}. Un interrupteur idéal a une résistance R_{ON} nulle et une résistance R_{OFF} infinie. En pratique, R_{ON} et R_{OFF} peuvent avoir des valeurs qui affectent la transparence d'un interrupteur au sein de la structure du circuit. Afin de minimiser leur influence sur les performances du circuit et les résultats des tests, il faut que R_{ON} soit la plus petite possible et que R_{OFF} soit la plus grande possible. Il faut également réduire les capacités parasites d'un interrupteur afin de minimiser leurs effets secondaires tel que le "feedthrough" d'horloge. Celui-ci peut être réduit en utilisant un interrupteur CMOS [85]. Il s'en suit, qu'afin de minimiser les effets secondaires d'un interrupteur, il faut qu'il soit soigneusement conçu. Dans la suite de cet exposé nous supposons que les interrupteurs considérés sont parfaits. L'analyse de leur influence sur la détectabilité des paramètres d'un circuit fera l'objet d'un travail futur.

Dans la section suivante nous allons illustrer la méthode de génération de tests à l'analyse de détectabilité de paramètres d'un autre circuit.

5.4 Analyse de détectabilité d'un amplificateur à large bande

Dans cette section nous allons analyser la détectabilité d'un circuit d'amplificateurs en cascade (figure 5.14) permettant d'obtenir une bande-passante assez large (50.8KHz) et un gain global du circuit de 101dB. Plus précisément, nous allons analyser tout d'abord, la détectabilité des paramètres du circuit en supposant que les gains des amplificateurs opérationnels sont infinis. Puis, nous entamons l'étude de l'influence d'un gain fini dépendamment et indépendamment de la fréquence, par rapport au cas idéal. Enfin, nous verrons qu'en utilisant les techniques DFT, on peut améliorer considérablement la détectabilité des différents paramètres du circuit.

Remarquons tout de suite que les résultats donnés dans les tableaux ci-après sont calculés pour une résolution de l'appareil de mesure de 0.01V et pour des tolérances de ±2% sur les paramètres et les gains finis des amplificateurs opérationnels (dans les cas où ceux-ci sont considérés comme des amplificateurs opérationnels non idéaux).

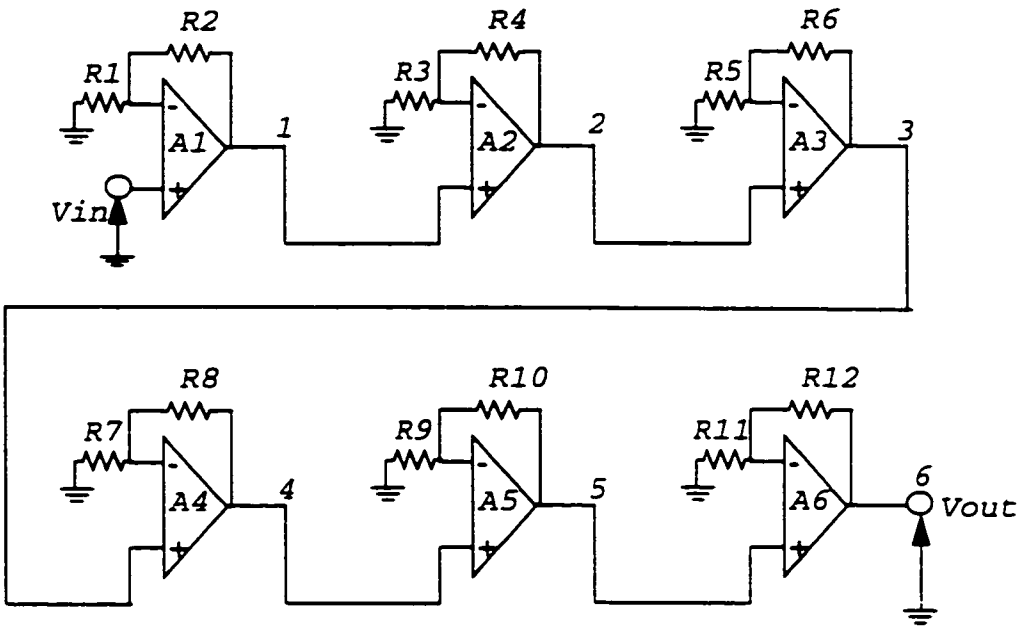


Figure 5.14: Amplificateur à large bande en cascade

5.4.1 Amplificateurs opérationnels de gain infini

Dans cette section les gains d'amplificateurs opérationnels sont supposés infinis. Supposons qu'on désire analyser la détectabilité des paramètres Ri, i=1,..., 12 en observant le gain à la sortie du circuit (noeud 6). Celui-ci a pour expression:

$$A_v = \left(1 + \frac{R_2}{R_1}\right)\left(1 + \frac{R_4}{R_3}\right)\left(1 + \frac{R_6}{R_5}\right)\left(1 + \frac{R_8}{R_7}\right)\left(1 + \frac{R_{10}}{R_9}\right)\left(1 + \frac{R_{12}}{R_{11}}\right) \qquad (7)$$

L'application de la méthode de génération de tests pour une fréquence de test fixé (inférieure à la fréquence de coupure du circuit), donne les résultats indiqués dans le tableau 5.4.

TABLEAU 5.4 : Résultats obtenus pour des gains d'amplis infinis.

paramètre	valeur nominale (Ω)	SOPD LOND [%]	détectabilité d [%]	temps CPU [s]
$^{*}R_{2n+1}$	5600	61.05 -36.12	56.05 71.56	131
$^{*}R_{2n+2}$	33000	56.51 -37.93	58.60 70.31	143
* n=0,1,...,5				

Voyons maintenant quel est l'impact d'un gain fini sur les détectabilités des paramètres.

5.4.2 Amplificateurs opérationnels de gain fini

Dans cette section nous allons analyser le cas où les gains des amplificateurs opérationnels sont finis (faisant abstraction, pour le moment, de leur variation en fonction de la fréquence), et voir leur influence sur la détectabilité des paramètres. Le circuit amplificateur de la figure 5.14 a été calculé en considérant des amplificateurs opérationnels de gains finis A_i (i=1,....6) de 100000 en boucle ouverte. L'expression générale du gain global (au noeud 6) du circuit s'écrit alors:

$$A_v = \left(\frac{A_1}{1+\beta_1 A_1}\right)\left(\frac{A_2}{1+\beta_2 A_2}\right)\left(\frac{A_3}{1+\beta_3 A_3}\right)\left(\frac{A_4}{1+\beta_4 A_4}\right)\left(\frac{A_5}{1+\beta_5 A_5}\right)\left(\frac{A_6}{1+\beta_6 A_6}\right) \quad (8)$$

avec:

$$\beta_{i+1} = \frac{R_{2i+1}}{R_{2i+1}+R_{2i+2}}, \; i = 0, 1, ..., 5$$

L'application de la méthode de génération de tests pour une fréquence de test fixée (inférieure à la fréquence de coupure du circuit), donne les résultats indiqués dans le tableau 5.5.

TABLEAU 5.5 : Résultats obtenus pour des gains d'amplis op. finis

paramètre	valeur nominale (Ω)	SOPD LOND [%]	détectabilité d [%]	temps CPU [s]
$^{*}R_{2n+1}$	5600	176.75 -58.69	18.03 57.36	77
$^{*}R_{2n+2}$	33000	142 -63.88	25.34 54.52	75
* n=0.1.....5				

D'après les tableaux 5.4 et 5.5, le fait de tenir compte d'un gain fini pour les amplificateurs opérationnels dégrade dans une importante proportion la détectabilité des paramètres du circuit.

Voyons maintenant l'impact d'un gain fini dépendant de la fréquence.

5.4.3 Amplis opérationnels de gain dépendant de la fréquence

Les amplificateurs opérationnels considérés ont une fréquence de transition $f_T = \frac{\omega_T}{2\pi}$de 1MHz, leur fonction de transfert est de la forme:

$$A_i(s) = \frac{A_i\omega_{0i}}{s+\omega_{0i}} = \frac{\omega_{Ti}}{s+\omega_{0i}} = \frac{10^5 20\pi}{s+20\pi}, \; i=1, \ldots, 6 \tag{9}$$

La fonction de transfert globale du circuit est obtenue en remplaçant les gains A_i, i=1,...,6, de l'équation 8 par leur valeur en fonction de la fréquence (équation 9). Elle a pour expression:

$$A_v(s) = \prod_{i=1}^{6}\left[\frac{\frac{A_i\omega_{0i}}{s+\omega_{0i}}}{1+\beta_i\frac{A_i\omega_{0i}}{s+\omega_{0i}}}\right] \tag{10}$$

avec

$$\beta_{j+1} = \frac{R_{2j+1}}{R_{2j+1}+R_{2j+2}}, \; j=0, 1, \ldots, 5 \tag{11}$$

L'équation 10 peut être transformée sous la forme:

$$A_v(s) = \prod_{i=1}^{6}\left[\frac{A_{vi}(0)}{\frac{s}{\omega_{Hi}}+1}\right] \tag{12}$$

avec:

$$A_{vi}(0) = \frac{A_i}{1+\beta_i A_i}, \text{ et } \omega_{Hi} = (1+\beta_i A_i)\omega_{0i}, \; i=1, \ldots, 6 \tag{13}$$

d'où:

$$|A_v(s)| = \frac{\prod_{i=1}^{6} A_{vi}(0)}{\sqrt{\prod_{i=1}^{6}\left[1+\left(\frac{\omega}{\omega_{Hi}}\right)^2\right]}} \qquad (14)$$

Les résultats de calcul des détectabilités des paramètres du circuit obtenus en observant le gain $|A_v(s)|$ (équation 14) sont montrés dans le tableau 5.6. La plage de fréquences considérée est $[1.10^6]Hz$.

TABLEAU 5.6 : Résultats obtenus pour des gains d'amplis op. finis dépendant de la fréquence.

paramètre	valeur nominale (Ω)	SOPD LOND [%]	détectabilité d [%]	Fr [Hz]	temps CPU [s] (pour 6 fréq.)
$^{*}R_{2n+1}$	5600	176.75 -58.69	18.03 57.36	1001	427
$^{*}R_{2n+2}$	33000	142 -63.88	25.34 54.52	1001	686
* n=0.1.....5					

Mise à part le temps de calcul, la comparaison des résultats des tableaux 5.5 et 5.6 montrent que les valeurs SOPD et LOND sont identiques. Cela provient du fait que la meilleure fréquence d'observation (1001Hz, tableau 5.6) se trouve dans la région où le gain global du circuit est constant en fonction de la fréquence. Rappelons que la bande du circuit considéré est de 50.8 KHz.

5.4.4 Application de la technique DFT

Les résultats trouvés jusqu'ici montrent que les détectabilités des paramètres du circuit ne sont pas satisfaisantes, plus particulièrement dans le cas où l'on tient

compte des gains finis des amplificateurs opérationnels. Voyons maintenant si l'on peut, grâce à la technique DFT, améliorer ces détectabilités. Supposons qu'on modifie le circuit (au moyen d'interrupteurs et d'un multiplexeur), de telle façon qu'on puisse contrôler et observer les noeuds 2 et 4 en plus des noeuds d'entrée et de sortie. Il s'en suit, que l'expression du gain global en fonction de la fréquence, de chacun des noeuds 2, 4 et 6 est de la forme:

$$|A_{v2}(s)| = \frac{\left(\frac{A_i}{1+\beta_i A_i}\right)\left(\frac{A_{i+1}}{1+\beta_{i+1} A_{i+1}}\right)}{\sqrt{\left[1+\left(\frac{\omega}{\omega_{Hi}}\right)^2\right]\left[1+\left(\frac{\omega}{\omega_{Hi+1}}\right)^2\right]}} \qquad (15)$$

avec i = 1 (noeud 2),3 (noeud 4),5(noeud 6)

Le tableau 5.7 montre les résultats de calcul des détectabilités des paramètres du circuit modifié (incluant les points de contrôle et d'observation). Ces résultats correspondent aux cas suivants: gains des amplificateurs opérationnels infinis, gains finis (sans tenir compte de la dépendance en fréquence), et gains dépendant de la fréquence.

TABLEAU 5.7 : Résultats obtenus avec le circuit modifié

paramètre	valeur nominale (Ω)	gains infinis			gains finis			gains dépendant de la fréquence			
		SOPD LOND [%]	d** [%]	temps CPU [s]	SOPD LOND [%]	d** [%]	temps CPU [s]	SOPD LOND [%]	d**	Fr [Hz]	temps CPU [s] (6 fréq.)
*R2n+1	5600	15.14 -13.01	87.91 89.76	10	26.76 -20.62	78.45 83.31	22	26.76 -20.62	78.45 83.31	1001 1001	655
*R2n+2	33000	14.93 -13.19	88.09 89.61	10	25.94 -21.14	79.08 82.89	22	25.94 -21.14	79.08 82.89	1001 1001	358

*n=0,....5.
** d: détectabilité

En comparant les résultats des tableaux 5.4, 5.5 et 5.7, nous constatons une fois de plus qu'en modifiant le circuit pour contrôler et observer les noeuds 2 et 4, on obtient une amélioration appréciable des détectabilités des paramètres du circuit. A titre d'exemple, dans le cas des amplificateurs opérationnels à gains finis la détectabilité des paramètres R_{2n+1} (n = 0,....,5) passe de 18.03% à 78.45% pour les déviations positives, et passe de 57.36% à 83.31% pour les déviations négatives.

Bien évidemment, si l'on modifie le circuit de telle manière à observer et à contrôler l'entrée et la sortie de chacun des six amplificateurs, on peut améliorer davantage les détectabilités des paramètres du circuit. Mais, pour tirer le meilleur profit des technique DFT, il faut trouver un bon compromis entre l'importance de la circuiterie à ajouter (*overhead*) et le niveau de détectabilité désiré.

5.5 Conclusion

Dans ce chapitre nous avons montré que la méthode de génération de tests proposée dans cette thèse peut être appliquée efficacement à l'analyse de détectabilité des circuits analogiques linéaires. Nous avons introduit une nouvelle mesure de détectabilité des déviations d'un paramètre, et avons définit les conditions de son indétectabilité. Nous avons également montré que les gains finis des amplificateurs opérationnels d'un circuit peuvent dégrader dans une bonne proportion les détectabilités des paramètres comparativement au cas idéal. Enfin, nous avons montré qu'en appliquant les techniques DFT, la détectabilité des défauts dans les circuits analogiques peut être considérablement améliorée. Cependant, il faut tenir compte des effets secondaires de la circuiterie ajoutée. Cela fera l'objet du travail futur.

Chapitre 6
Conclusion Générale

Jusqu'à nos jours, le test des circuits analogiques le plus répandu dans l'industrie est un test de vérification des spécifications. Dans ce genre de test, en plus d'être coûteux, les circuits sont sur-testés ou sous-testés puisqu'il n'y a pas de mesure quantitative de qualité permettant de disposer d'un critère d'arrêt. Il en découle qu'il faut développer des nouvelles techniques faisant appel aux modèles de défaut, qui permettent de minimiser le coût du test de production. Or, la mise en oeuvre de telles techniques se heurtent à plusieurs difficultés. Ces difficultés émanent de l'immensité de l'espace de recherche provenant de la continuité des grandeurs physiques qui caractérisent les circuits analogiques, et de l'incertitude associée à ces grandeurs qui est causée par les tolérances des paramètres. En outre, la non-linéarité de ces caractéristiques rend le problème de test analogique plus complexe.

Dans cette thèse, nous avons mis au point une méthode de génération de tests pour détecter les défauts catastrophiques et paramétriques dans les circuits analogiques linéaires. Nous avons formulé le problème de génération de tests comme une série de problèmes d'optimisation. En général, ce sont des problèmes multidimensionnels non-linéaires très complexe et très difficiles résoudre mettant en oeuvre beaucoup de temps de calcul.

Initialement nous avons résolu ces problèmes grâce à une méthode de programmation non-linéaire, à savoir la programmation quadratique séquentielle (SQP), disponible dans l'outil de mathématiques MATLAB. Cependant, de telles méthodes ne garantissent pas que la solution trouvée est globale. Cela peut donc conduire à une mauvaise sélection de tests générés. En outre, elles ne sont pas automatiques et dépendent de la précision de certains paramètres introduits par l'usager. Nous avons alors élaboré une méthode basée sur la programmation logique à contraintes (CLP) qui permet de résoudre les problèmes d'optimisation en question comme une série de problèmes de satisfaction à contraintes. Cette méthode trouve les optimums (minimum et maximum) globaux d'une fonction multivariables non-linéaire, en les cernant à l'intérieur des bornes étroites, infaillibles et garanties. Cette qualité des bornes trouve son origine dans les propriétés de l'arithmétique des intervalles sur laquelle repose la programmation logique à contraintes (CL)). Parmi ces propriétés figurent l'arrondissement vers l'extérieur des résultats, le processus de rétrécissement des intervalles et son exactitude, ainsi que la propriété d'inclusion.

Afin d'illustrer l'efficacité de la méthode proposée, nous l'avons appliquée à des fonctions mathématiques non-linéaires reconnues comme difficiles. Les résultats obtenus sont meilleures ou similaires à ceux générés par d'autres méthodes. Ensuite nous l'avons appliquée à des circuits électriques dans une perspective de génération de tests pour ces circuits. La comparaison des résultats générés dans le cas du filtre

biquadratique avec ceux obtenus à l'aide de la méthode SQP concordent dans une certaine mesure dans presque tous les cas, à l'exception d'un cas où l'optimum calculé par SQP est un minimum local. Cela est incorrect et peut donc conduire à une sélection erronée de tests.

La méthode proposée peut être utilisée pour résoudre le problème d'analyse de tolérance des performances d'un circuit dans le pire des cas, analyse qui joue un rôle important dans la vérification de design. Ce problème est efficacement traité dans le cadre de cette thèse, comme une composante du problème de génération de tests.

Bien que le temps CPU moyen par optimisation obtenu pour les circuits considérés, est raisonnable nous avons suggéré certaines techniques pour accélérer davantage l'exécution des algorithmes proposés. Parmi ces techniques figurent l'usage du test de monotonicité conjugué avec une forme améliorée de la valeur moyenne (une des formes d'une fonction intervalle extension).

Nous avons aussi montré dans le cadre de cette thèse que la qualité du procédé de fabrication ainsi que la résolution de l'appareil de mesure peuvent considérablement affecter la détectabilité des paramètres du circuit sous-test, et par suite affecter la qualité des tests générés. Nous avons également montré que les gains finis des amplificateurs opérationnels peuvent dégrader sérieusement la détectabilité des paramètres par comparaison au cas idéal. Enfin, nous avons vu qu'il est possible d'augmenter dans une importante proportion la détectabilité des paramètres, en pratiquant du

design orienté testabilité. Cependant, il faut tenir compte des effets secondaires et des coûts additionnels de la circuiterie ajoutée.

Il est important de signaler que la méthode de génération de tests proposée peut être améliorée davantage, par des considérations pratiques tel que la réduction des espaces de recherche des paramètres en établissant des valeurs seuils à partir desquelles on décide de pratiquer du design orienté testabilité.

Enfin, il faut mentionner que les concepts de cette méthode peuvent être étendus pour couvrir le test des circuits analogiques non-linéaires.

BIBLIOGRAPHIE

[1] ABDERRAHMAN A., CERNY E. et KAMINSKA B. (1995). *Effective Test Generation for Analog Circuits*. Proc. of International Mixed Signal Testing Workshop, Grenoble, France, 86-91.

[2] ABDERRAHMAN A., CERNY E. et KAMINSKA B. (1996). *Optimization-based Multifrequency Test Generation for Analog Circuits*. Journal of Electronic Testing (JETTA), vol. 9,no 1/2, 59-73.

[3] ABDERRAHMAN A., CERNY E. et KAMINSKA B. (1997). *CLP-based Multi-frequency Test Generation for Analog Circuits*. 15th IEEE VLSI Test Symposium, 158-165.

[4] ABDERRAHMAN A., CERNY E. et KAMINSKA B. (1997). *Worst-Case Tolerance Analysis and CLP-based Multifrequency Test Generation for Analog Circuits*. Submitted to Journal of IEEE Transactions on Computer-Aided Design.

[5] ABRAMOVICI M., BREUER M. A. et FRIEDMAN A.D. (1990). *Digital Systems Testing and Testable Design*. Computer Science Press.

[6] BANDLER J.W. et SALAMA A. E. (1985). *Fault Diagnosis of Analog Circuits*. Proceedings of the IEEE, vol. 73, 1279-1324.

[7] WALKER A., ALEXANDER W. E. et LALA P. K. (1992). *Fault Diagnosis in Analog Circuits Using Element Modulation*. IEEE Design & Test of Computers, vol. 9, 19-29.

[8] MILOR L. et VISVANATHAN V. (1989). *Detection of Catastrophic Faults in Analog Integrated Circuits*. IEEE Transactions on Computer-Aided Design, vol. 8, N0 2, 114-130.

[9] WILSON Q. F. et DAY D. B. (1987). *Practical Automatic Test Program Generation Constraints*. Proceedings of Automatic Test Conference and Workshop.

[10] MALY W., STROJWAS et DIRECTOR S. W. (1986). *VLSI Yield Prediction and Estimation: A Unified Framework*. IEEE Transactions on Computer-Aided Design, vol. CAD-5, 114-130.

[11] ISMAIL M. et FIEZ T. (1994). *Analog VLSI: Signal and Information Processing*", McGraw-Hill.

[12] SEBEKE C., TEIXEIRA J. P. et OHLETZ M.J. (1995). *Automatic Extraction and Simulation of Layout Realistic Faults for Integrated Analogue Circuits*. Proc. of European Design and Test Conference, 464-468.

[13] WALKER D. et DIRECTOR S. W. (1986). *VLASIC: A Catastrophic Fault Yield Simulator for Integrated Circuits*. IEEE Transactions on CAD, vol. 5, no 4, 541-556.

[14] WALKER D. (1990). *VLASIC System User Manual Release 1.3*. Carnegie-Mellon Univ. Pittsburg, U.S.A.

[15] SHEN J.P., MALY W. et FERGUSON F.J. (1985). *Inductive Fault Analysis of MOS Integrated Circuits*. IEEE Design and Test of Computers, vol. 2, no 6, 13-26.

[16] MEIXNER A. et MALY W. (1991). *Fault Modeling for the Testing of Mixed Integrated Circuits*. Internatinal Test Conference, 564-572.

[17] ARENDSEN R.G.J., BRULS E.M.J.G et HERMANN O.E. (1996). *Exploiting the Modeling Flexibility of Mixed-Mode Mixed-Level Simulation for Efficient Fault Simulation of Large Mixed-Signal Circuits*. Proc. Int. Mixed Signal Testing Workshop, Quebec, Canada, 5-10.

[18] SACHDEV M. et ATZEMA B. (1995). *Industrial Relevance of Analog IFA: A Fact or a Fiction*. Proc. Int. Test Conf., 10-14.

[19] BELL I.M. et SPINKS S.J. (1995). *Analog Fault Simulation for the Structural Approach to Analogue and Mixed-Signal IC Testing*. Proc. inter. Mixed Signal Testing Workshop, Grenoble, France, 10-14.

[20] HARVEY R.J.A. et al. (1994). *Analogue Fault Simulation Based on Layout Dependent Fault Models*. Proc. Int. Test Conf, 641-649.

[21] PAN C.Y. et al. (1994). *A Comprehensive Fault Macro-Model for Opamps*. Proc. ICCAD, 344-348.

[22] ZWOLINSKI M., CHALK C. et WILKINS B.R. (1996). *Analogue Fault Modeling and Simulation for Supply Current Monitoring*. Proc. European Design and Test Conference, 547-552.

[23] NAGI N. et ABRAHAM J.A. (1992). *Hierarchical Fault Modeling for Analog and Mixed-Signal Circuits*. IEEE VLSI Test Symposium, 96-101.

[24] BELL I.M. et al. (1995). *Fault Oriented Test and Fault Simulation of Mixed Signal Integrated Circuits*. Proc. of IEEE Int. Symposium on Circuits and Systems, vol. 1, Seattle, Washington.

[25] POVAZANEG J., VOLEK T. et TAYLOR G. (1995). *Analog Test in Frequency and Time Domains*", Proc. inter. Mixed Signal Testing Workshop, Grenoble, France, 239-243.

[26] SAVARIA Y. (1988). *Conception et Vérification des Circuits VLSI*", Editions de l'Ecole Polytechnique de Montreal.

[27] BERKOWITZ R.S. (1962). *Conditions for Network-Element-Value Solvability*. IEEE Transactions on Circuits Theory, vol. CT-9, 24-29.

[28] SAEKS R. (1977). *A Measure of Testability and its Apllication to Test Point Selection Theory*. 20th Midwest Symposium on Circuits and Systems, Texas Tech. Univ. Lublock.

[29] HEMINK J.G.J. et al. (1990). *Testability Analysis of Analog Systems*", IEEE Transactions on Computer-Aided Design, vol.9, no 6, 573-583.

[30] SLAMANI M. (1994). "*Test Intégrè, Diagnostic et Analyse de la Testabilité dans les Circuits Intégrès Analogiques basés sur le concept de la Sensibilité*", Thèse de doctorat, École Polytechnique de Montréal.

[31] WAGNER K. et WILLIAMS T. (1989). *Design for Testability of Analog/Digital Networks*. IEEE Transactions on Industrial Electronics, vol. 36, 227-230.

[32] WAGNER K. et WILLIAMS T. (1988). *Design for Testability of Mixed-Signal Integrated Circuits*. International Test Conference, 823-828.

[33] WEY C.L. (1990). *Built-in Self-Test (BIST) Structure for Analog Circuit Fault Diagnosis*. IEEE Transactions on Instrumentation and Measurement, vol. 39, no 3, 517-521.

[34] SLAMANI M. et KAMINSKA B. (1994). *An Integrated Approach for Analog Circuit Testing with a Minimum Number of Detected Parameters*. International Test Conference, 631-639.

[35] CHATTERJEE et al. (1996). *DC Built-in Self-test for Linear Analog Circuits*. IEEE Design & Test of Computer, 26-33.

[36] ARABI K. et KAMINSKA B. (1996). *Oscillation Test Strategy for Analog and Mixed-Signal Integrated Circuits*. Proc. Of the IEEE Test Symposium, 476-482.

[37] SAKAGUCHI K. et KANEKO M. (1991). *Fault Diagnosis with Short-Circuit for Linear Analog Networks*. 1991 IEEE International Symposium on Circuits and Systems, 2068-2071.

[38] YOU Z. et al. (1995). *Analog System-Level Fault Diagnosis Based on Symbolic Method in Frequency domain*. IEEE Transactions on Instrumentation and Measurement, 28-35.

[39] LIN P.U. et ELCHERIF Y.S. (1985). *Analog Circuits Fault Dictionnary: New Approaches and Implementation*. Int. J. of Circuit Theory and Applications, vol. 13, 149-172.

[40] SACHDEV M. (1995). *A Realistic Defect Oriented Testability Methodology for Analog Circuits*. Journal of Electronic Testing and Applications (JETTA), vol 6, no 3, 265-276.

[41] JOHNSON A.T. (1979). *Efficient Fault Analysis in Linear Analog Circuits*. IEEE Transactions on Circuits and Systems, CAS-26, no 7, 475-484.

[42] SALAMA A.E. et AMER F.Z. (1991). *Parameter Identification Approach to Fault Diagnosis of Switched Capacitor Circuits*. IEE Proc. of Electronic Circuits and Systems, vol. 139, 467-472.

[43] SLAMANI M. et KAMINSKA B. (1992). *Analog Circuit Fault Diagnosis Based on Sensitivity Computation and Functional Testing*. IEEE Design & Test of Computers, 30-39.

[44] HATZOPOULOS A.A. et KONTOLEON J.M. (1987). *Efficient Fault Diagnosis in Analogue Circuits Using a Branch Decomposition Approach*. IEE Proc. of Electronic Circuits and Systems, vol. 134, 149-157.

[45] LIU R.W. (1987). *Selected Papers on Analog Fault Diagnosis*. IEEE Press, New York.

[46] LU Y. et DANDAPANI R. (1993). *Hard Fault Diagnosis in Analog Circuits Using Sensitivity Analysis*. IEEE VLSI Test Symposium, 225-229.

[47] Semiconductor Industry Technology Workshop Conclusions (1993). Semiconductor Industry Association.

[48]European Design and Test Conference (1996). Panel discussion on DFT.

[49] SOMA M. (1993). *Fault Coverage of DC Parametric Tests for Embeded Analog Amplifiers*. IEEE International Test Conference, 566-573.

[50] MILOR L. et VISVANATHAN V. (1989). *Detection of catastrophic faults in analog integrated circuits*. IEEE trans. Computer-Aided Design, vol.8, 114-130.

[51] MILOR L. et SANGIOVANNI-VINCENTLLI (1990). *Optimal Test Set Design for Analog Circuits*. Proc. ICCAD, 294-297.

[52] MILOR L. et SANGIOVANNI-VINCENTLLI (1994). *Minimizing Production Test Time to Detect Faults in Analog Circuits*. IEEE trans. Computer-Aided Design, vol.13, No.6, 796-813.

[53] SHEN-JEN T. (1990). *Test Vector Generation for Linear Analog Devices*. Proc. IEEE International Test Conference, 592-597.

[54] BEN HAMIDA N. et KAMINSKA B. (1993). *Analog Circuit Testing Based on Sensitivity Computation and New Circuit Modeling*. Proc. IEEE International Test Conference, 652-661.

[55] BEN HAMIDA N., SAAB K., MARCHE D., KAMINSKA et QUESNEL G. (1996). *LIMSoft: Automated Tool for Design and Test Integration of Analog Circuits*. IEEE International Test Conference, 571-580.

[56] NAGI N., CHATTERJEE A., BALIVADA A. et ABRAHAM J. (1993). *Fault-based Automatic Test Generator for Linear Analog Circuits*", Proc. of ICCAD, 88-91.

[57] DEVARAYANADURG G. et SOMA M. (1994). *Analytical Fault Modeling and Static Test Generation for Analog ICs*. Proc. of ICCAD, 44-47.

[58] BALIVADA A., CHEN J. et ABRAHAM J.A. (1996). *Analog Testing with Time Response Parameters*. IEEE Design & Test of Computers, 18-25.

[59] CAMPLIN D. A. et al. (1991). *Can Supply Monitoring be Applied to the Testing of Analogue and Digital Portions of Mixed ASICs?* Proc. of IEEE International Test Conference, 538-542.

[60] ECKERSALL K. R. et al. (1993). *Testing Mixed Signal ASICs through the Use of Supply Current Monitoring*. Proc. of European Test Conference, 385-391.

[61] RICHARDSON A. et al. (1995). *The Application of IDDX Test Strategies in Analogue and Mixed Signal IC's*. Proc. of IEEE International Mixed Signal Testing Workshop, 206-211.

[62] MIURA Y. (1996). *Real Time Current Testing for A/D converters*. IEEE Design & Test of Computers, 34-41.

[63] SMAVAJULA S. S. et al. (1996). *Analog Fault Diagnosis Based on Ramping Power Supply Current Signature Clsuters*. IEEE Transactions on Circuits and Systems-II, vol. 43, no 10, 703-712.

[64] SODEN J.M et HAWKINS C.F. (1996). *I_{DDQ} Testing: Issues Present and Future*. IEEE Design & Test of Computers, 61-65.

[65] PAN C. Y. et CHENG K. T. (1995). *Pseudo-random Testing and Signature Analysis for Mixed-signal Circuits*", IEEE International Conference on VLSI Design.

[66] Optimization Toolbox User's Guide (1992). MathWorks, Inc.

[67] SCHITTOWSKI K. (1985). *NLQPL: A Fortran-subroutine Solving Constrained Nonlinear Programming Problems*. Operations Research, Vol. 5, 485-500.

[68] BANDLER J. W. , LIU P. C. et CHEN J. H. K. (1975). *Worst Case Network Tolerance Optimization*. IEEE Trans. Microwave Theory Tech., vol. MTT-23, 630-641.

[69] H. TROMP (1977). *The generalized Tolerance Problem and Worst Case Search*. Proc. IEEE Conf. Computer-Aided-Design of Electronic and Microwave Circuits and Systems, 72-77.

[70] JAFFAR J. et MAHER M. (1994). *Constraint Logic Programming: A Survey*. Journal of Logic Programming, 19/20, 503-581.

[71] MOORE R.E. (1966). *Interval Analysis*. Prentice Hall.

[72] MOORE R.E. (1979). *Methods and Applications of Interval Analysis*. Philadelphia, SIAM.

[73] CLEARY J.G. (1987). *Logical Arithmetic*. Future Computing Systems, Vol. 2.No 2, 125-149.

[74] OLDER W.J. (1995). *Introduction to Constraint Logic Programming*. Tutorial given at Inter-University Center in Computer Architecture and VLSI, Université de Montréal.

[75] OLDER W.J. et VELLINO A. (1993). *Constraint Arithmetic on Real Intervals, in Contraint Logic Programming: Selected Research*. Colmerauer and Benhamou, MIT Press.

[76] MACKWORTH A.K. (1977). *Consistency in Network of Relations*. Artificial Intelligence 8, 99-118.

[77] SKELBOE S. (1979). *True Worst-Case Analysis of linear electrical circuits by interval arithmetic*. IEEE Transactions on Circuits and Systems, Vol. CAS-26, 874-879.

[78] CLP(BNR) Prolog User Guide and Reference Manual (1988).

[79] HASSEN E. (1980). *Global Optimization Using Interval Analysis: The Multi_Dimensional Case*. Numerische MathematiK, Vol. 34, 247-270.

[80] ASAITHAMBI N. S., SHEN Z. et MOORE R.E. (1982). *On Computing the Range of Values*. Computing, Vol. 28, 225-237.

[81] MOORE R.E. et RATSCHEK H. (1987). *Inclusion Functions and Global Optimization II*. Mathematical Programming.

[82] KOLEV L. V. et AL. (1988). *Interval Mathematics Algorithms for Tolerance Analysis*. IEEE Transaction on Circuits and Systems, Vol. 35, No 8, 967-975.

[83] BAINTER J.R. (1975). *Active Filter Has Stable Notch, and Response can be Regulated.* Electronics, 115-117.

[84] SKELBOE S. (1974). Computation of Rational Interval functions. BIT 14, 87-95.

[85] PHILIP E. A et DOUGLAS R. H. (1987). *CMOS Analog Circuit Design.* Holt, Rinehart and Winston.