Optimisation du test de production de circuits analogiques et RF par des techniques de modélisation statistique

N. Akkouche

UNIVERSITÉ DE GRENOBLE

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Micro et Nano Electronique**

Arrêté ministériel : 7 août 2006

Présentée par

Nourredine AKKOUCHE

Thèse dirigée par **Salvador MIR**
et codirigée par **Emmanuel SIMEU**

Préparée au sein du **Laboratoire TIMA**
dans **l'École Doctorale Electronique, Electrotechnique, Automatique et Traitement du Signal (E.E.A.T.S)**

Optimisation du test de production de circuits analogiques et RF par des techniques de modélisation statistique

Thèse soutenue publiquement le **Vendredi 9 Septembre 2011**,
devant le jury composé de :

M. Olivier GAUDOIN
Professeur à Grenoble INP, Président
M. Patrick GIRARD
DR-CNRS à Montpellier, Rapporteur
M. Fabrice MONTEIRO
Professeur à l'Université Paul Verlaine-Metz, Rapporteur
M. Ahcène BOUNCEUR
Maître de conférence à l'Université Européenne de Bretagne, Examinateur
M. Salvador MIR
DR-CNRS à Grenoble, Directeur de thèse
M. Emmanuel SIMEU
Maître de conférence à l'Université Joseph Fourier, Co-Directeur de thèse
M. Thierry FALQUE
Ingénieur à ST Microelectronics de Grenoble, Invité

A la mémoire de mon père
A ma chère mère

A mes frères : Tahar, Hacène, Boukhalpha, Kamal et Samir
A mes sœurs : Nadia et Zoubida

A ceux et celles qui m'ont aidés

⁂ *Remerciements* ⁂

Je tiens à exprimer ici toute ma gratitude à mon directeur de thèse, M. Salvador Mir, *Directeur de recherche au CNRS de Grenoble et responsable du groupe RMS*, de m'avoir accueilli au sein de son équipe, pour tous les précieux conseils qu'il m'a donnés et pour le temps qu'il a consacré pour diriger cette thèse.

Je remercie chaleureusement mon co-directeur de thèse, M. Emmanuel Simeu, *Maître de conférences à l'Université Joseph Fourier*, pour ses conseils et discussions très constructifs, ayant permis une très bonne orientation des travaux de cette thèse.

Je voudrais exprimer mes remerciements les plus sincères à M. Olivier Gaudoin, *Professeur à Grenoble INP*, d'avoir accepté de présider le jury de cette thèse.

Mes remerciements chaleureux s'adressent également à M. Patrick Girard, *Directeur de recherche au CNRS de Montpellier*, et M. Fabrice Monteiro, *Professeur à l'Université Paul Verlaine-Metz*, d'avoir accepté de lire ce manuscrit en tant que rapporteurs.

Mes remerciements vont aussi à M. Ahcène Bounceur, *Maître de conférences à l'Université Européenne de Bretagne*, pour toute l'aide qu'il ma apporté et d'avoir accepté d'examiner cette thèse. Je remercie également M. Thierry Falque, *Ingénieur à ST Microelectronics de Grenoble* pour sa participation au jury de cette thèse.

Enfin, je tiens aussi à remercier mes collègues de bureau, l'équipe du groupe RMS et les membres du laboratoire TIMA, qui m'ont bien accueillis et m'ont aidés à accomplir mon travail de thèse.

Je ne peux oublier de remercier tous les membres de ma famille pour leurs soutiens et leurs encouragements.

Table des matières

Table des figures

Liste des tableaux

Chapitre 1

Introduction

1.1 Contexte et motivations

La taille des appareils électroniques qui nous entourent ne cesse de diminuer. Des premiers ordinateurs qui occupaient des hangars jusqu'aux téléphones portables d'aujourd'hui qui peuvent être intégrés dans une montre, la tendance a toujours été à la miniaturisation et à l'intégration. L'intégration des technologies micro-électroniques permet actuellement de fabriquer des dispositifs incluant des parties ou des blocs de nature hétérogène. La miniaturisation a permis l'intégration dans un même appareil de plusieurs fonctionnalités. Par exemple, un téléphone portable sert à téléphoner, prendre des photos et des vidéos, écouter de la musique, se situer géographiquement avec des fonctions de géo-localisation, accéder à internet, ... Mais est-ce que ces produits deviennent plus fiables, avec l'augmentation du niveau d'intégration ? Il est primordial d'accompagner cette tendance à l'intégration et à la miniaturisation par une plus grande fiabilité et sûreté de fonctionnement. Le test des circuits électroniques peut intervenir à différents stades de la vie d'un circuit, de la conception jusqu'à l'utilisation dans l'application finale en passant par les différentes phases de production.

Le test fonctionnel consiste à vérifier que le circuit fonctionne correctement, c'est-à-dire à vérifier si toutes ses spécifications, imposées par le cahier des charges, sont respectées. Un autre type de test est le test structurel qui consiste à vérifier directement l'existence ou non d'un défaut sur un composant du circuit (résistance, transistor, ...). Les tests structurels sont générés à partir de simulations sur un modèle théorique du circuit dans lequel on injecte des modèles de fautes. Toutefois, ce test est très dépendant du modèle de fautes utilisé. Dans le cas des circuits analogiques, mixtes et Radio Fréquence (RF), le manque de modèles de fautes performants a limité le succès des techniques de test structurel. Ainsi, les circuits intégrés avec un comportement analogique sont testés par des tests fonctionnels explicites : on mesure directement l'ensemble des performances puis on compare ces mesures aux spécifications définies par le concepteur du circuit sous test. Ainsi, le circuit est classé comme fonctionnel si toutes ses spécifications sont satisfaites,

autrement, il est classé défaillant. Cependant, cette démarche est très coûteuse en temps et en équipements de test.

Le rôle de plus en plus important joué par le test sur le coût des circuits analogiques et mixtes, impose une optimisation des méthodes de test pour réduire le coût de production des circuits intégrés. Comme le temps de test est un facteur déterminant de ce coût, il est évident qu'il faut ordonner les tests de manière à détecter au plus tôt les circuits défectueux. Ou bien recourir au test d'un nombre réduit de performances, qui sera effectué dans un temps raisonnable, tout en acceptant un risque d'erreur de test minimum.

1.2 Contributions

L'objectif de ce travail est de proposer des méthodes d'optimisation du test des circuits analogiques et radio fréquences. Ces méthodes doivent satisfaire à deux exigences primordiales :

1. généralité : être applicable à tous les types de circuits quel que soit le nombre et la nature des tests ;
2. adaptabilité : travailler avec les données disponibles et pouvoir exploiter des données obtenues au cours du test.

Le premier critère exige de la méthode d'optimisation des tests de traiter tous les circuits quel que soit le nombre des tests nécessaires. La méthode proposée doit optimiser le test d'un circuit simple nécessitant une dizaine de tests jusqu'à des circuits plus complexes nécessitants plusieurs centaines de tests. Il faut aussi tenir compte de la nature des tests, la méthode doit traiter aussi bien des circuits dont tous les tests ont le même comportement que des circuits avec des tests ayant des comportements différents. En effet, il est plus facile de modéliser un circuit où tous les tests ont un même comportement qu'un autre circuit où chacun des tests nécessaires a un comportement différent. La deuxième exigence impose une flexibilité des méthodes proposées en fonction de la quantité et de la nature des données disponibles. Lors de la conception du circuit, les seules données disponibles sont issues de la simulation Monte Carlo, qui ne comporte souvent que des circuits fonctionnels. Alors qu'après la production et les premiers lots de circuits testés, on dispose de grandes quantités de circuits fonctionnels et des circuits défectueux. Or, nous avons plus besoin de données sur les circuits défectueux que sur les circuits fonctionnels pour quantifier l'erreur commise durant le test.

Nous proposons une méthode d'ordonnancement des tests basée sur la modélisation statistique des tests d'un circuit. Cette méthode utilise la modélisation statistique pour générer un échantillon de données de plusieurs millions d'instances pour pallier au manque de données dans la phase de conception, au temps de simulation qui est important et pour satisfaire à l'exigence de précision de l'ordre du ppm (partie par million) dans l'estimation des métriques de test. Ainsi, le modèle construit va capturer le comportement

paramétrique du circuit sous test et permettra de générer des circuits défectueux avec des déviations paramétriques. Selon la nature des tests d'un circuit, la modélisation paramétrique par une loi multinormale sera utilisée si tous les tests ont un comportement multinormal ou bien par un modèle basé sur les copules. Si aucun des deux modèles paramétriques (multinormal et copule) ne modélisent le comportement du circuit, un modèle non paramétrique plus général sera utilisé. En revanche, il est moins précis que les deux modèles précédents. Une fois le modèle statistique validé, le rééchantillonnage du modèle permet de générer un échantillon de plusieurs millions d'instances du circuit sous test. Cet échantillon est utilisé pour une estimation au ppm près du taux de défauts (proportion des circuits défectueux passant les tests). La minimisation de cette erreur (taux de défauts) par différents algorithmes de recherche (méthode de séparation et d'évaluation, algorithmes génétiques et méthodes de recherche flottante), suivant la dimension du circuit, permettra d'ordonner les tests d'un circuit de telle manière à mettre au début les tests qui détectent le plus de circuits défectueux.

La méthode d'ordonnancement des tests a été implémentée selon trois variantes en fonction de la nature des données disponibles :

1. la méthode d'ordonnancement des tests utilise un petit échantillon de circuits fonctionnels (pas de circuits défectueux) ;
2. la méthode de sélection utilise uniquement les données issues des circuits défectueux (pas de circuits fonctionnels) ;
3. la méthode de sélection et d'ordonnancement utilise à la fois des circuits fonctionnels et défectueux.

Ces méthodes sont représenté sur la Figure 1.1.

1.3 Structure de la thèse

La thèse est structurée en 5 chapitres avec une introduction, une conclusion et des perspectives.

Dans le chapitre 2, nous présenterons les concepts de base du test analogique et radio fréquences à travers la définition des tests structurels, fonctionnels et alternatifs. Après quelques rappels de définitions concernant le vocabulaire du test, nous allons aborder le test structurel dépendant de la modélisation et la simulation de fautes. Ce test est ensuite comparé au test fonctionnel, traité dans ce travail, ainsi qu'au test alternatif. On finira, par introduire les métriques de test.

Un état de l'art des méthodes d'ordonnancement et de réduction des tests est présenté dans le chapitre 3. Une classification des méthodes en 4 classes est proposée. Cette classification est basée sur les techniques utilisées dans l'ordonnancement ou la réduction du nombre de tests. Chacune des classes est illustrée à travers l'exposé de plusieurs méthodes

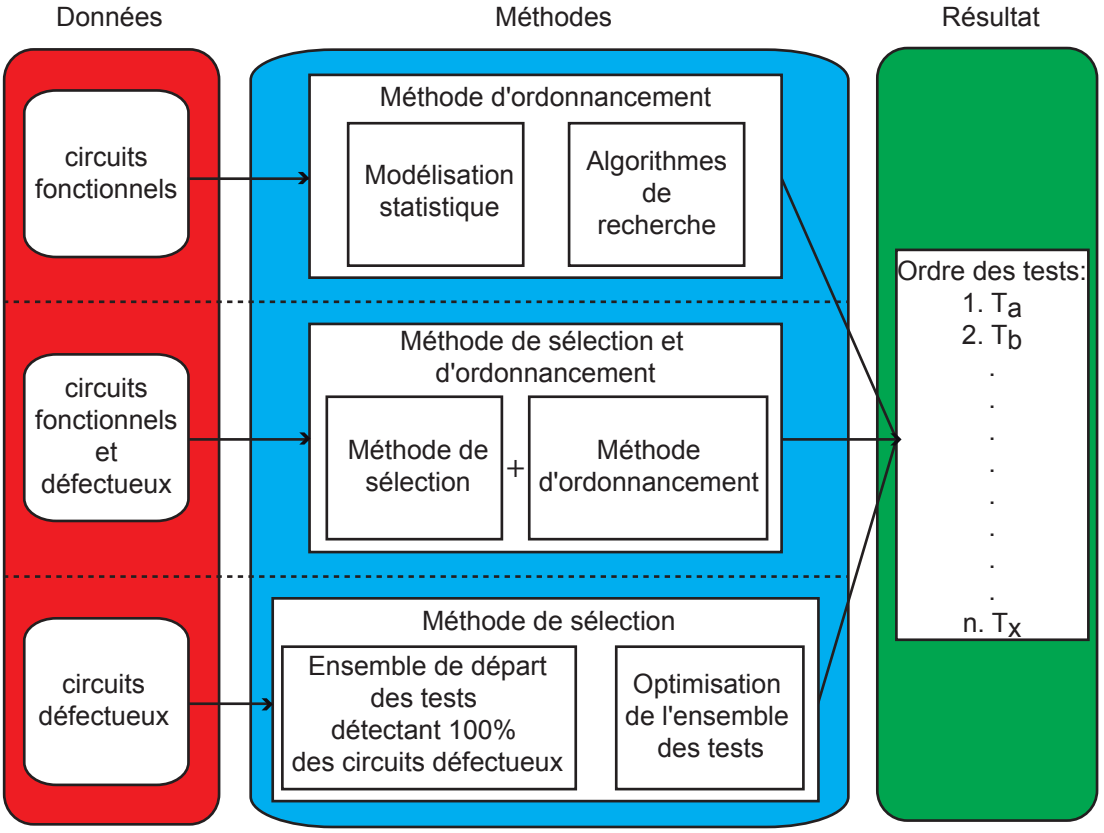


Figure 1.1 – La méthode d'ordonnancement des tests et ses variantes.

employant des techniques comme la corrélation, les métriques de test, l'identification des paramètres ou la classification.

Les différents modèles statistiques (multinormal, copule et non paramétrique) sont exposés dans le chapitre 4. Chaque modèle est défini puis illustré par une application sur un circuit. Une fois le modèle validé, un échantillon de plusieurs millions d'instances est généré. Cet échantillon est utilisé par les algorithmes de recherche : méthode de séparation et d'évaluation, algorithmes génétiques et méthodes de recherche flottante suivant la complexité du circuit sous test. Ces méthodes feront l'objet du chapitre 5 et seront illustrées avec des exemples d'applications.

Le chapitre 6 sera consacré aux résultats expérimentaux issus de l'application de la méthode d'ordonnancement des tests sur un circuit industriel. Ce circuit conçu par IBM est composé de 143 tests classés en différentes catégories : numériques, analogiques et radio fréquences. Les différentes variantes de la méthode d'ordonnancement seront mises à l'épreuve car on dispose à la fois de données sur les circuits fonctionnels et défectueux.

Une conclusion et quelques perspectives de recherche seront présentées à la fin de ce manuscrit de thèse.

Chapitre 2

Concepts de base du test analogique et RF

2.1 Introduction

Un récapitulatif des travaux de recherche publiés avant 1979 sur le test des circuits analogiques est proposé dans [1]. A l'époque, le problème du test des circuits analogiques se posait pour tester les circuits imprimés qui commençaient à devenir de plus en plus complexes grâce entre autre à l'évolution des circuits MSI [1] (intégration à échelle moyenne) et LSI [2] (intégration à grande échelle). Depuis les années 1980, on est capable de réaliser des circuits appelés circuits VLSI [3] (intégration à très grande échelle) pouvant abriter des centaines de millions de composants. Le test de cette nouvelle génération de circuit est très développé dans le cas des circuits numériques mais pose de nombreux problèmes dans le cas des circuits analogiques.

Aujourd'hui, avec la forte demande de circuits analogiques et mixtes, et la complexité croissante des circuits, beaucoup de travaux de recherche se développent dans tous les domaines du test analogique, allant de la modélisation de fautes jusqu'à la conception en vue du test.

Dans ce chapitre nous ferons un tour d'horizon de l'environnement du test mixte (analogique et numérique) et RF à travers les concepts de fautes et de test. Dans la première section de ce chapitre, nous donnerons quelques définitions des différents termes utilisés dans le domaine du test des circuits analogiques. Dans les autres parties du chapitre, nous nous intéresserons aux différents travaux publiés dans les domaines suivants :

- Modélisation et simulation de fautes.
- Types de test.

1. Medium Scale Integrated circuits
2. Large Scale Integrated circuits
3. Very Large Scale Integrated circuits

2.2 Définitions

Comme il n'existe pas de terminologie standard pour les termes utilisés dans le domaine du test analogique, et pour faciliter la lecture de ce document, voici les définitions des termes importants utilisés :

Défaut : défaut physique qui affecte la réalisation d'un circuit.

Faute : modélisation d'un défaut physique dans le but de simuler l'effet de ce dernier sur le circuit.

Faute catastrophique : modélisation d'un défaut majeur comme un court-circuit ou un circuit ouvert.

Faute paramétrique : modélisation des fluctuations de l'environnement de fabrication qui engendrent des variations sur les sorties du circuit.

Diagnostic : détermination de la cause du dysfonctionnement d'un circuit.

Optimisation d'un ensemble de tests : réduction du nombre de vecteurs de test d'un ensemble, tout en détectant les mêmes fautes que l'ensemble de départ. Le but de l'optimisation des tests est de réduire le temps nécessaire à l'application de l'ensemble des tests sur des équipements de test très coûteux, et réduire ainsi le coût du test de production.

Couverture de fautes : le rapport du nombre de fautes détectées par rapport au nombre de fautes globales. La couverture de fautes dépend du modèle de fautes utilisé.

Paramètres process : les paramètres physiques du circuit (résistance, capacité, dimensions d'un transistor, ...).

Paramètres design : appelés aussi les performances. Représentent les paramètres permettant de décider si le circuit est fonctionnel ou non.

Paramètres de test : appelés aussi critères de test, ils peuvent être une partie des paramètres du design ou bien d'autres paramètres pouvant aider à décider si le circuit est fonctionnel ou non.

Simulation de type Monte Carlo : elle consiste à générer un grand nombre d'instances d'un circuit en faisant varier de façon pseudo-aléatoire tous ses paramètres de design. Si le nombre d'instances est suffisamment important, on peut ainsi considérer que l'on couvre toute la plage des variations possibles. Cette méthode est sans doute la plus précise, mais elle est extrêmement coûteuse en temps de simulation.

2.3 Types de fautes

L'étude des défauts physiques susceptibles d'être présents dans les circuits intégrés nécessite une modélisation. Un modèle de fautes permet de représenter les défauts physiques qui peuvent affecter les masques (layout) d'un circuit pour pouvoir simuler leurs

conséquences sur le comportement du circuit. Un bon modèle de fautes doit être simple à utiliser et doit représenter fidèlement les effets des défauts physiques du circuit. La qualité d'un ensemble de tests est déterminée par la couverture de fautes et le modèle de fautes utilisé.

Le modèle de fautes le plus utilisé pour les circuits numériques est le modèle des collages [4]. La puissance du modèle des collages réside dans sa simplicité d'utilisation et dans sa capacité à détecter la majorité des défauts physiques. En effet même si beaucoup de défauts ne peuvent être représentés par le modèle des collages, ils peuvent être détectés par ce modèle [2]. Les fautes dans les circuits intégrés analogiques peuvent être classées en deux catégories :

- Les fautes catastrophiques ;
- Les fautes paramétriques.

Avant de détailler les deux types de fautes, nous analysons le processus de production des circuits micro-électroniques.

2.3.1 Processus de fabrication des circuits micro-électroniques

Le processus de fabrication des circuits intégrés utilise des réactions chimiques et des procédés physiques (Figure 2.1) qui sont très dépendants de l'environnement, très sensibles aux variations de température, aux vibrations et aux différentes variations des outils de fabrication. Des petites fluctuations de l'environnement de fabrication peuvent affecter un circuit micro-électronique et créer des fautes catastrophiques ou paramétriques.

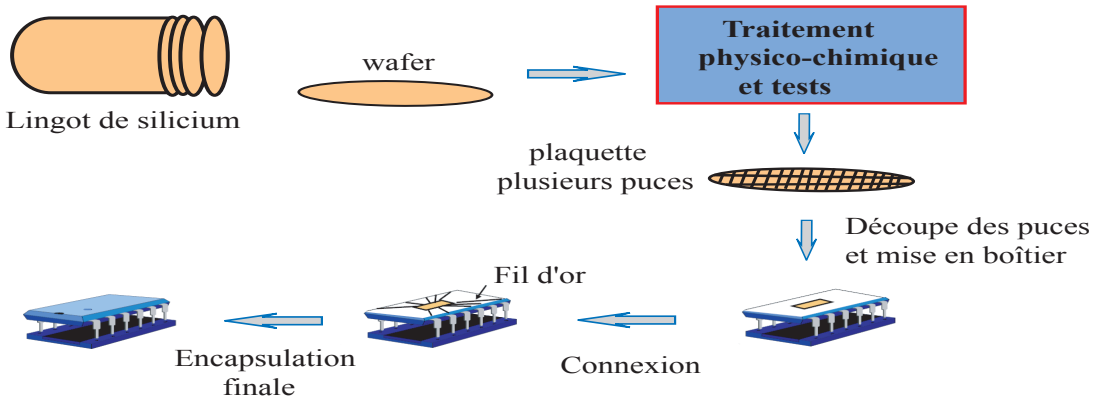


Figure 2.1 – Les étapes de production.

Les étapes de production d'un circuit dépendent de la technologie employée, les principales phases de production sont résumées à travers les étapes suivantes :

- Les fondeurs extraient du sable le silicium. Il est ensuite chauffé et purifié. Toute impureté risque ici de réduire l'efficacité de la puce. Une fois chauffé, le silicium est

4. Stuck at Fault

découpé en plaquettes ou wafers, de 10 à 30 cm de diamètre et de 0.5 mm d'épaisseur. Ce sont sur ces plaquettes que seront imprimés les schémas électroniques des circuits.
- La gravure représente l'étape la plus délicate. Armée du composant, la plaquette de silicium, et des schémas du circuit, la chaîne de fabrication commence par chauffer à plus de 1 000 °C la plaquette de silicium. Par une réaction naturelle va se former une couche d'oxyde à la surface de la plaquette, qui servira de protection à la plaquette de silicium pure.
- La deuxième étape du processus de gravure passe par l'ajout d'une couche de vernis photosensible. Dès lors, il suffit de passer l'ensemble sous des rayons UV en appliquant le masque défini par le schéma du circuit comme modèle. Après cette étape, la plaquette de silicium est encore une fois nettoyée de ses impuretés. Vient alors l'étape du dopage, qui consiste en un bombardement d'ions de manière à charger positivement ou négativement certaines zones de la future puce.
- Lorsque le dopage est terminé, la gravure recommence mais, cette fois-ci, sur une autre couche de la puce, afin d'établir les connexions entre les différents composants électroniques d'un même circuit.
- Lorsque l'étape de gravure se termine, les puces d'une plaquette de silicium passent un premier test, celui du microscope à balayage automatique. Ce premier test sert à détecter les principales anomalies qui pourraient affecter une puce. Un deuxième test paramétrique s'applique sur l'ensemble des circuits d'une plaquette. Ce n'est que lorsque ces deux tests sont passés que les puces sont déclarées aptes à la vente. Elles sont alors découpées et insérées dans un boîtier. Ce boîtier protège en outre la puce contre l'humidité, la corrosion et la contamination atmosphérique.

2.3.2 Les fautes catastrophiques

La définition des fautes catastrophiques diffère d'un auteur à un autre. Pour certains auteurs, les fautes catastrophiques sont des fautes qui correspondent à des défauts aléatoires localisés dans un point [5]. Par exemple, une particule de poussière sur un masque photolithographique entraînant des déformations locales qui peuvent engendrer des court-circuits et des circuits ouverts. En revanche, pour d'autres auteurs les fautes catastrophiques sont des fautes qui engendrent un fonctionnement du circuit complètement différent du fonctionnement normal, même si l'origine de cette faute n'est qu'une petite variation d'un paramètre du circuit. Dans la suite de cette thèse, nous avons adopté la première définition car elle permet de différencier les types des fautes non pas par rapport au fonctionnement du circuit mais suivant l'origine de la faute. En outre, pour les circuits analogiques, il n'est pas facile de définir la limite entre une petite déviation et une grande déviation pour pouvoir classer sans ambiguïté une faute selon la deuxième définition.

5. Spot Defect

2.3.3 Les fautes paramétriques

Comme pour les fautes catastrophiques, pour certains auteurs, les fautes paramétriques sont des fautes dues aux fluctuations des paramètres du processus de fabrication qui en général n'engendrent pas un comportement complètement différent du circuit mais causent des déviations des sorties du circuit qui sont en dehors des intervalles de tolérance. Pour d'autres auteurs, les fautes paramétriques sont les fautes qui engendrent des déviations des sorties du circuit en dehors des intervalles de tolérance. En utilisant la deuxième définition, cela reviendrait à dire que même si l'origine de la faute est un court-circuit dû à une particule de poussière, si le comportement du circuit diffère seulement de son comportement initial mais reste globalement le même, alors la faute est considérée comme paramétrique. Pour les mêmes raisons que précédemment, nous utiliserons la première définition pour la suite du document.

Comme les fautes paramétriques engendrent des déviations des paramètres de sortie du circuit et que ces déviations peuvent être plus au moins grandes suivant le paramètre considéré, il est donc plus difficile de tester ces fautes. En effet, il ne suffit pas de trouver des vecteurs de test qui activent les fautes, mais il faut aussi trouver les meilleurs paramètres qui permettent d'avoir des déviations en sortie du circuit en dehors des intervalles de tolérance.

2.4 Modélisation de fautes

Un modèle de fautes représentatif des défauts réels et simple à utiliser est fondamental pour développer une stratégie de test efficace. En effet, la mesure de l'efficacité d'un ensemble de vecteurs de test en termes de défauts réels détectés est basée sur le modèle de fautes utilisé. Si pour un ensemble de tests donné, on a un taux de couverture de fautes de 100%, cela ne veut pas dire qu'on va détecter tous les défauts physiques. L'efficacité d'un ensemble de tests dépend aussi de la représentativité du modèle de fautes utilisé. Plus le modèle de fautes est représentatif de la majorité des défauts physiques, plus on aura de défauts détectés. En général, comme pour les circuits numériques, toutes les modélisations supposent que si la faute existe alors elle est unique. Cependant certaines techniques prennent en compte le cas des fautes multiples, mais elles sont rarement applicables aux circuits actuels car beaucoup trop complexes.

Étant donné que les fautes catastrophiques engendrent un fonctionnement complètement différent du circuit, elles sont plus faciles à détecter. En effet comme la plupart des fautes catastrophiques affectent tous les paramètres du circuit, le problème du choix des meilleurs paramètres à mesurer en sortie ne se pose pas. En général, un simple test DC ou un test en courant IDDQ[6] peut détecter la majorité des fautes catastrophiques [3]. Le test en courant IDDQ est basé sur la mesure du courant d'alimentation du circuit. En

6. Quiescent current

général pour les fautes catastrophiques le courant d'alimentation mesuré est différent du courant d'alimentation du circuit correct. Par contre, les fautes paramétriques engendrent des déviations des paramètres de sortie du circuit et ces déviations peuvent être plus au moins grandes suivant le paramètre considéré. Il est donc plus difficile de tester ces fautes. En effet, il ne suffit pas de trouver les vecteurs de test qui activent la faute, mais il faut aussi trouver les meilleurs paramètres qui permettent d'avoir une déviation en sortie du circuit en dehors de la plage de tolérance acceptable.

2.4.1 Modélisation au niveau composant ou structurel

La modélisation au niveau composant consiste à modéliser les défauts au niveau structurel. Pour un transistor MOS [7] (Figure 2.2), le modèle de fautes catastrophique comprend six fautes possibles, les trois courts-circuits entre les trois terminaux du transistor (Grille, Source et Drain) et les trois circuits ouverts sur ces mêmes terminaux [4]. Mais dans la pratique, les fautes les plus probables sont les courts-circuits et les circuits ouverts sur le drain et la source [5].

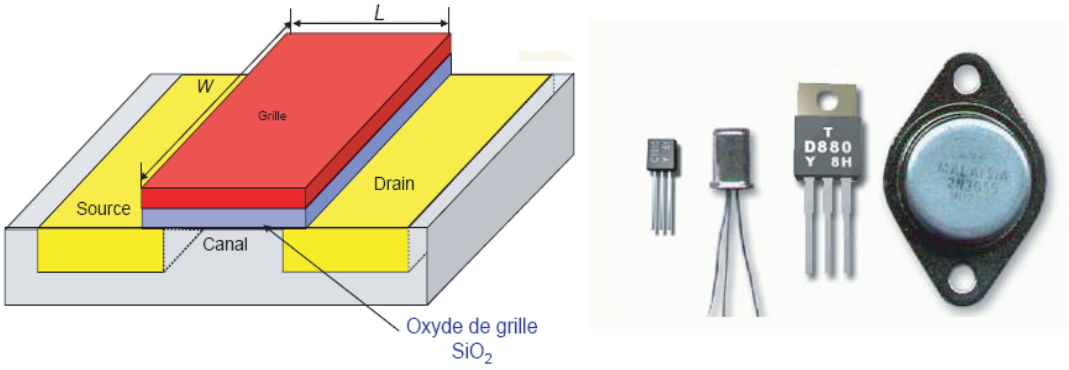


Figure 2.2 – Le transistor MOS.

Pour les composants passifs (résistances et capacités), nous avons 2 fautes catastrophiques par composant, un court-circuit qui est modélisé par une résistance de faible valeur en parallèle avec le composant et un circuit ouvert modélisé par une résistance de valeur très élevée en série avec le composant.

2.4.2 Modélisation au niveau fonctionnel

Le modèle fonctionnel des fautes permet de modéliser les effets des défauts physiques des blocs analogiques au niveau fonctionnel. Pour la modélisation au niveau fonctionnel, on divise le circuit en plusieurs modules fonctionnels. Pour chaque module et chaque faute

7. Metal Oxide Semiconductor

du module on effectue des simulations analogiques au niveau structurel pour modéliser les effets de la faute au niveau fonctionnel. Après abstraction des fautes au niveau fonctionnel, les modules fonctionnels sont représentés comme des boîtes noires caractérisées par les différences entre les sorties du bon circuit et des circuits avec fautes [6].

2.5 Simulation de fautes et génération des vecteurs de test

2.5.1 Simulation de fautes

Le but d'un simulateur de fautes est de déterminer les effets des défauts sur le comportement du circuit et d'évaluer la qualité des jeux de vecteurs de test en calculant le taux de couverture obtenu selon un modèle de fautes donné. Les simulateurs de fautes pour les circuits numériques utilisent des techniques connues comme la simulation de fautes parallèles ou la simulation de fautes concurrentes qui exploitent le fait qu'on utilise des simulations logiques et que les différences entre le bon circuit et le circuit fautif sont minimes. Ces techniques permettent de réduire considérablement le temps nécessaire à la simulation de toutes les fautes. Malheureusement, ces techniques sont difficilement utilisables pour les circuits analogiques [7].

L'approche la plus utilisée pour la simulation de fautes des circuits analogiques est basée sur l'utilisation d'un simulateur électrique comme SPICE et ELDO. La méthode consiste à exécuter les étapes suivantes :

- simulation du circuit sans faute ;
- introduction d'une faute dans le circuit ;
- simulation du circuit avec faute ;
- comparaison des résultats des deux simulations.

Les trois dernières étapes sont répétées pour chaque faute.

2.5.2 Injection des fautes

Les fautes paramétriques sont modélisées par des variations de valeurs des paramètres des composants (comme la valeur de la résistance ou de la capacité) en dehors de leur intervalle de tolérance. Par contre, pour les transistors, nous avons plusieurs paramètres du processus de fabrication qui interviennent dans la modélisation du transistor. Comme pour la simulation des circuits analogiques, on utilise des modèles de transistors définis par plusieurs paramètres, les fautes paramétriques sont modélisées par des déviations en-dehors des intervalles de tolérance des paramètres les plus importants du modèle utilisé. Ces paramètres peuvent être : la longueur L, la largeur W, la tension de seuil V_{th} et le gain en courant β. Même si on suppose que les distributions des paramètres du processus de

fabrication sont normales, les distributions des paramètres du modèle de transistor utilisé pour la simulation ne sont pas forcement normales. En effet, les paramètres du modèle de transistor sont des combinaisons des paramètres du processus de fabrication et ces combinaisons peuvent être non linéaires. Les distributions statistiques et les intervalles de tolérance des paramètres du modèle du transistor sont en général donnés par le fabricant.

2.5.3 Génération automatique des vecteurs de test

Le but de la génération automatique des vecteurs de test (ATPG [8]) est d'obtenir un ensemble minimal de vecteurs de test qui permet de détecter le maximum de fautes. Pour les circuits numériques, il existe plusieurs algorithmes qui permettent de générer des vecteurs de test de façon déterministe selon un modèle de fautes donné. La majeure difficulté des ATPGs pour les circuits numériques réside dans la complexité importante des circuits (plusieurs millions de transistors et plus d'une centaine d'entrées-sorties). Par contre pour les circuits analogiques, nous avons des circuits beaucoup plus petits en nombre de composants et d'entrées-sorties, mais dont la spécificité ne permet pas d'utiliser des algorithmes de génération automatique comme dans le cas des circuits numériques.

Jusqu'à ces dernières années les vecteurs de test pour les circuits analogiques et mixtes sont générés manuellement. En général, les concepteurs utilisent les vecteurs fonctionnels pour le test structurel, ce qui fait que les circuits analogiques sont soit sous-testés soit sur-testés, ce qui augmente considérablement le coût du test. Par ailleurs, la croissance des circuits analogiques ces dernières années a rendu de plus en plus difficile la génération manuelle des vecteurs de test. C'est pourquoi plusieurs travaux commencent à apparaître dans le domaine de la génération des vecteurs de test dédiés aux circuits analogiques et mixtes, mais leur maturité reste très inférieure à celle atteinte par la génération des vecteurs de test pour les circuits numériques.

La génération automatique de vecteurs de test pour les circuits analogiques peut être décomposée en plusieurs domaines : la génération pour le test structurel, l'optimisation des vecteurs de test, la génération compatible avec un testeur.

2.5.4 Optimisation automatique des vecteurs de test

La formulation du problème d'optimisation des vecteurs de test consiste en : soit un ensemble de vecteurs de test donné et un ensemble de fautes à détecter, le coût du test dépend du nombre de vecteurs de test à appliquer et de l'ordre dans lequel ces vecteurs sont appliqués. En effet, plus on a de vecteurs de test à appliquer, plus le temps de test est important. Comme le test d'un circuit fautif est arrêté dès que la faute est détectée, il est avantageux en termes de temps de test, de placer les vecteurs qui détectent le plus de fautes au début du test. Le but des techniques d'optimisation des tests est de réduire le

8. Automatic Test Pattern Generation

nombre de vecteurs dans l'ensemble de test initial et de classer les vecteurs sélectionnés pour réduire le coût du test tout en conservant la même couverture de fautes que celle de l'ensemble de départ [8].

2.6 Types de test

Le test d'un circuit a pour objectif la détection de problèmes liés à la fabrication ou au vieillissement, mais pas à la conception du circuit. En fin de fabrication, comme pour tous les tests ultérieurs, le circuit est supposé exempt d'erreur de conception.

Le test d'un circuit intégré s'effectue en différentes phases, destinées à vérifier des caractéristiques variées. A la fin de la fabrication, le test peut classiquement être découpé en trois phases [9] :

Test des motifs de surveillance du process : ces motifs de validation sont positionnés sur la plaquette servant de substrat (tranche de silicium dans la plupart des cas), entre les circuits à fabriquer. Ils sont définis par le fabricant et sont indépendants du circuit proprement dit. Ils permettent, en fin de fabrication, de s'assurer que les étapes technologiques du procédé de fabrication se sont déroulées correctement et que les dispositifs intégrés (transistors, interconnexions, ...) ont les caractéristiques attendues. Il s'agit donc essentiellement d'un test électrique de caractérisation (mesure de résistances, de caractéristiques courant-tension, ...).

Test des circuits avant découpe : quand les étapes de fabrication sont validées, un premier test des circuits est réalisé sur la plaquette avant sa découpe. Ce test a pour but d'éviter le montage en boîtier de circuits grossièrement défectueux, le montage et le boîtier pouvant avoir un coût très élevé par rapport au circuit nu.

Test des circuits en boîtier : une fois la plaquette découpée et les circuits montés en boîtier, les tests précédents sont refaits à fréquence plus élevée et largement complétés.

2.6.1 Test hors ligne et en ligne

Les différentes phases de test indiquées précédemment correspondent au test de fin de fabrication, effectué lorsque le circuit n'est pas encore placé dans son environnement opérationnel. Un tel test est dit «*hors ligne*».

Par opposition, un test «*en ligne*» est un test exécuté par le circuit, alors qu'il est connecté dans son environnement opérationnel et que l'application est en cours d'exécution.

2.6.2 Test fonctionnel, structurel et alternatif

Que le test de la fonction du circuit soit effectué en ligne ou hors ligne, il peut utiliser une approche fonctionnelle, structurelle ou alternative.

L'approche fonctionnelle consiste à définir des vecteurs de test permettant de couvrir tous les modes de fonctionnement possibles du circuit, tels qu'ils sont spécifiés dans la fiche technique (ou le cahier des charges).

L'approche structurelle consiste à vérifier la structure interne du circuit et le bon fonctionnement des éléments de base. L'approche structurelle est aujourd'hui la plus utilisée pour le test de fin de fabrication, mais elle peut dans certains cas être couplée avec une séquence de vérification fonctionnelle, indépendante de tout modèle de fautes. Ceci peut permettre un gain de temps de développement et même une réduction du nombre total de vecteurs de test, si les vecteurs fonctionnels utilisés comme base du programme de test sont bien choisis. Une telle génération «*mixte*» permet aussi, dans certains cas de circuits séquentiels complexes, de contourner les limitations des outils de génération de vecteurs de test (ATPG), en mettant à profit la connaissance du circuit par le concepteur.

L'approche alternative a été récemment proposé pour les systèmes ou les circuits analogiques, mixtes et RF[9] (nécessitant un test fonctionnel) [10, 11, 12, 13]. Les raisons étant que le temps de test des performances est très long, en plus du coût très élevé des équipements ATE[10] utilisés. En particulier pour le cas des circuits RF, le nombre des performances à vérifier est très important. Dans cette approche, les performances du circuit sous test ne sont pas directement mesurées en utilisant les méthodes conventionnelles, mais par prédiction à partir d'un ensemble réduit de mesures de test des valeurs des performances. Cette analyse est généralement effectuée en utilisant les techniques de régression statistique (en particulier, la régression multiple non linéaire pour le cas des circuits RF). Le test alternatif est basé sur l'hypothèse que les variations des performances ainsi que celles des mesures de test dépendent des variations des paramètres physiques du circuit sous test. D'où la possibilité de prédire les variations, et notamment les valeurs, des performances à partir de celles des mesures de test. La Figure 2.3 résume ce principe.

9. Radio Frequency
10. Automatic Test Equipment

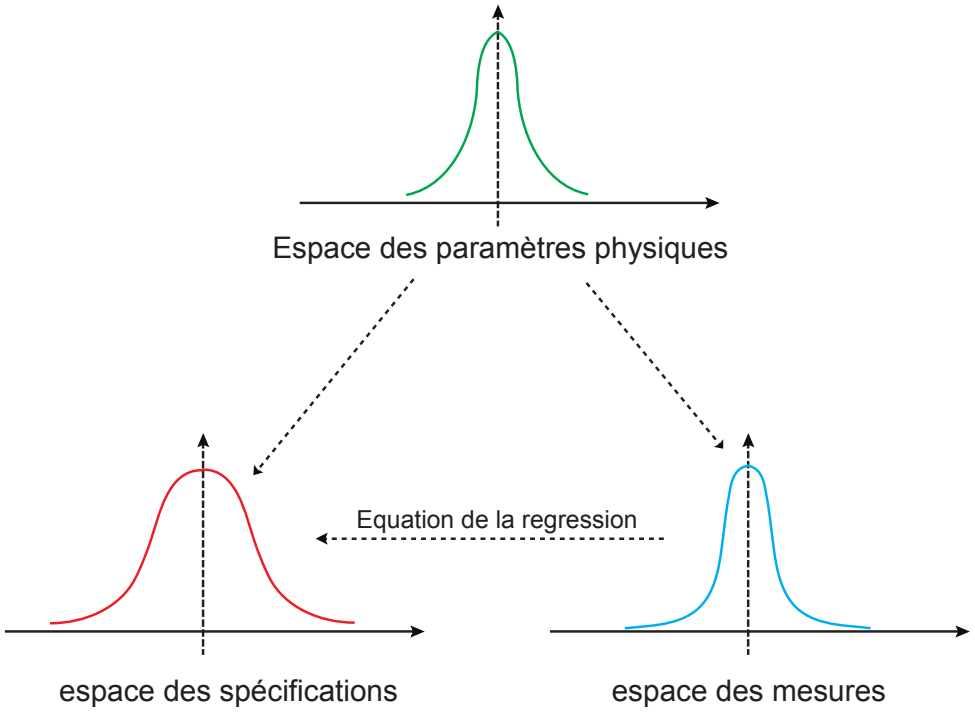


Figure 2.3 – Principe d'un test alternatif.

2.6.3 Test externe et autotest

La réalisation la plus classique d'un test consiste à connecter le circuit à une machine chargée d'appliquer des valeurs sur les broches d'entrées, puis de comparer les valeurs qui apparaissent sur les broches de sortie à des valeurs prédéfinies.

Cet équipement de «*test externe*» peut avoir différentes caractéristiques, selon qu'il s'agit de tester le circuit sous pointes, de vérifier la fonction d'un circuit après encapsulation, ou encore de réaliser un test paramétrique. De façon générale, cet équipement de test automatique ATE (Figure 2.4) est coûteux et ne permet pas souvent de tester le circuit à sa fréquence nominale de fonctionnement.

Une autre approche consiste à limiter au strict minimum le matériel externe nécessaire pendant le test. Dans ce cas, des éléments sont implantés directement dans le circuit afin de permettre son test.

Naturellement, pour que l'information soit utilisable, il faut s'assurer que les dispositifs «d'*autotest intégrés*», ou «*BIST*[11]» sont capables de se tester eux-mêmes.

Les dispositifs de BIST peuvent permettre de limiter le matériel externe à l'alimentation et à la génération des horloges. Ils peuvent aussi n'être utilisés que pour tester une partie du circuit ; dans ce cas, un ATE complet reste nécessaire, mais sa complexité peut être réduite, limitant d'autant son coût.

Un autre avantage du BIST est la possibilité de réaliser le test à la fréquence nominale de fonctionnement du circuit, puisque les éléments de test sont intégrés dans le circuit et

11. Built-In Self-Test

Figure 2.4 – Équipement de test automatique (ATE).

peuvent donc fonctionner à la même vitesse que les éléments logiques réalisant la fonction. Ceci permet de vérifier simplement le bon fonctionnement dynamique du circuit, ce qui prend une importance cruciale avec l'évolution des technologies et l'augmentation des fréquences d'horloge.

2.6.4 Test déterministe, aléatoire et pseudo-aléatoire

Que le test soit réalisé par des dispositifs de BIST ou par un équipement externe, des valeurs d'entrée doivent être appliqués sur le circuit. Ces valeurs peuvent être obtenues selon une approche fonctionnelle (pour vérifier la réalisation d'une liste de fonctions) ou par un algorithme permettant d'assurer la détection d'un ensemble de fautes donné. Nous parlerons dans ce cas de «*test déterministe*», au sens où la séquence des valeurs d'entrée appliquées au circuit est définie de façon à détecter une liste précise de problèmes potentiels.

Une autre approche consiste à envoyer sur les entrées du circuit des valeurs choisies *aléatoirement*. Ceci réduit notablement le temps passé à déterminer ces valeurs et il est possible de calculer statistiquement une longueur minimum de séquence à appliquer pour obtenir un certain niveau de qualité de test. Il est cependant indispensable de pouvoir comparer les sorties obtenues à des valeurs de référence, correspondant à un circuit fonctionnant de manière nominale.

L'approche aléatoire est aussi une approche utilisée pour l'implantation de dispositifs de BIST. Toutefois, un vrai générateur de valeurs aléatoires est difficile à réaliser ; un BIST aléatoire utilise donc classiquement une séquence de valeurs d'entrée figée, mais qui possède certaines propriétés statistiques permettant de réaliser le test selon le principe du test aléatoire. Un tel test est appelé «*pseudo-aléatoire*».

Le test pseudo-aléatoire garde l'inconvénient d'une séquence longue à appliquer, par rapport à une séquence déterministe. Dans la pratique, une grande partie des fautes dé-

tectées par un test déterministe peut souvent être détectée avec le début d'une séquence pseudo-aléatoire. Ensuite, un nombre réduit de fautes nécessite d'allonger considérablement la séquence. Certains concepteurs utilisent donc une approche «*mixte*», dans laquelle une séquence pseudo-aléatoire réduite est appliquée, suivie d'une séquence déterministe permettant de détecter les dernières fautes potentielles.

2.6.5 Test avec ou sans compaction des résultats

Lors de l'application d'une séquence de test, les sorties du circuit doivent être observées à chaque cycle et être comparées avec des valeurs de référence :

- dans le cas d'un test externe, cela signifie qu'il faut stocker dans la mémoire de l'ATE les valeurs attendues sur toutes les sorties pendant toute la durée du test, ce qui entraîne un coût au niveau de l'achat de l'équipement ;
- dans le cas d'un autotest, il faudrait stocker toutes ces valeurs dans une mémoire interne au circuit, ce qui n'est généralement pas réalisable sans induire un coût prohibitif en surface.

Pour pallier ce problème, les résultats du test peuvent être compactés en une donnée représentative appelée signature ; on parle alors de test «*compact*». Dans ce cas, les sorties du circuit sont connectées en entrée d'un bloc de compaction et une nouvelle valeur de signature est générée à chaque cycle en fonction de ces sorties et éventuellement de la valeur précédente de la signature.

La compaction peut être spatiale (réduction du nombre de bits à comparer) ou temporelle (comparaison effectuée seulement à certains cycles). Le plus souvent, la compaction est à la fois spatiale et temporelle et seule la dernière signature, obtenue à la fin du test, est comparée avec une référence ; seule cette référence a donc à être stockée dans le circuit, ce qui induit un coût très faible.

Naturellement, la génération de la signature entraîne une perte d'information. Il peut donc exister des cas de masquage d'erreur où plusieurs sorties erronées, dans le même cycle ou dans des cycles distincts, se compensent pour donner une signature juste à la fin du test. La fonction de compaction doit être choisie de façon à limiter la probabilité d'un tel masquage.

2.7 Conception en vue du test ou pour la testabilité

La testabilité peut être définie, de manière générale, comme l'aptitude d'un circuit ou d'un système à être testé. Cette aptitude dépend de deux notions complémentaires, fondamentales en test [9] :

- la contrôlabilité de chaque nœud électrique interne, c'est-à-dire le degré de facilité avec lequel chaque nœud peut être forcé à chacun des niveaux électriques, en

positionnant des niveaux adaptés sur les seules entrées primaires du circuit ou du système ;
- l'observabilité de chaque nœud électrique interne, c'est-à-dire le degré de facilité avec lequel le niveau électrique de chaque nœud peut être déterminé à partir de la seule lecture des sorties primaires du circuit ou du système.

Dans le cas d'un circuit complexe, il est devenu quasi-incontournable, même lorsque les meilleurs choix possibles sont faits lors des différentes étapes de la conception, d'ajouter au circuit certaines fonctionnalités dictées par le niveau de testabilité requis et non pas par les fonctions opérationnelles que le circuit doit réaliser. Cette «*conception en vue du test*» (DFT [12]) est généralement fondée sur un partitionnement du circuit en blocs de natures diverses, et sur l'application de méthodes adaptées à chaque type de bloc.

2.7.1 Calcul des métriques de test

Les métriques de test sont des mesures pour évaluer la qualité d'une technique de test. Dans le cas des fautes catastrophiques et des circuits numériques le paramètre le plus utilisé est la *Couverture de Fautes F* qui désigne la probabilité de détection des circuits avec faute. Cette probabilité est estimée comme suit :

$$F = \frac{\text{Le nombre de fautes qui sont détectées}}{\text{Le nombre total de fautes}} \tag{2.1}$$

La probabilité de détecter une faute dans un circuit numérique est égale soit à 1 (faute totalement détectée) ou à 0 (faute non détectée). Par contre pour les circuits analogiques, cette probabilité prend des valeurs différentes pour chaque faute et qui appartient à l'intervalle [0,1]. En plus, les fautes dans les circuits analogiques peuvent aussi bien être le résultat d'une déviation d'un seul paramètre que celui de déviations de plusieurs paramètres à la fois.

Il est important à noter qu'à elle seul, une bonne valeur de la couverture de fautes ne garantit pas une bonne technique de test pour les circuits analogiques, car il se peut que la technique détecte tous les circuits défaillants, tout en rejetant beaucoup de circuits fonctionnels. Donc, pour évaluer correctement la qualité d'un test, il faut évaluer un certain ensemble de métriques de test.

Dans ce qui suit nous définissons quelques métriques de test, en plus de la couverture de fautes, qui permettent d'évaluer d'une manière significative la qualité d'un test.

- Le Rendement (Y) : c'est la proportion des circuits fonctionnels, il est donné comme suit :

$$Y = \mathbf{P}(\text{Circuit est fonctionnel}).$$

12. Design For Testability

– Le Rendement de Test (Y_T) : c'est la proportion des circuits qui passent le test, il est donné comme suit :

$$Y_T = \mathbf{P}(\text{Circuit passe le test}).$$

– La Couverture de Rendement (Y_C) (respectivement la Perte de Rendement (Y_L)) : c'est la proportion des circuits qui passent (respectivement qui échouent) le test parmi les circuits fonctionnels. Ces proportions sont données comme suit :

$$Y_C = \mathbf{P}(\text{Circuit passe le test/il est fonctionnel})$$

$$Y_L = \mathbf{P}(\text{Circuit échoue au test/il est fonctionnel}) = 1 - Y_C.$$

– Taux de Défauts (D) : c'est la proportion des circuits défaillants parmi ceux qui passent le test, il est donné comme suit :

$$D = \mathbf{P}(\text{Circuit est défaillant/il passe le test}).$$

Ces métriques sont suffisantes pour avoir l'information nécessaire sur la qualité d'un test. D'autres métriques typiquement utilisées incluent :

– La Fausse Acceptation (FA) : appelée aussi *erreur de type I*. C'est une autre appellation du taux de défauts.
– Le Faux Rejet (FR) : appelé aussi *erreur de type II*. Il représente la proportion des circuits fonctionnels parmi ceux qui échouent le test, il est donné comme suit :

$$FR = \mathbf{P}(\text{Circuit est fonctionnel/il échoue le test}).$$

Ces métriques de test peuvent être calculées pour des fautes catastrophiques et paramétriques. Il existe des relations entre ces métriques, notamment entre la couverture de fautes et le taux de défauts. Cette relation, appelée formule de *Williams et Brown*, a été définie pour la première fois par [14] où les fautes ont été considérées comme équiprobables (chaque faute a la même probabilité d'occurrence). Ensuite, d'autres relations à base de celle-ci ont été développées afin de considérer le cas des fautes non-équiprobables (catastrophiques [15] et paramétriques [16]).

2.7.1.1 Cas des fautes catastrophiques

Dans le cas des fautes catastrophiques équiprobables, la couverture de fautes se calcule en utilisant la formule classique (2.1). Dans [14], une première relation entre le Taux de

défauts et la Couverture de fautes, appelée formule de *Williams et Brown*, a été définie comme suit :

$$D = 1 - Y^{1-T} \tag{2.2}$$

avec D le taux de défauts, Y le rendement et $T = F$ la couverture de fautes donnée par (2.1).

En analysant le layout d'un circuit, on note que les fautes ne peuvent pas être équiprobables. D'où la couverture de fautes sous l'hypothèse de la non-équiprobabilité des fautes définie par [15] sous le nom de *Couverture de Fautes Pondérée*.

$$T = \Omega = \frac{\ln \prod_{j=1}^{m}(1 - p_j)}{\ln \prod_{i=1}^{n}(1 - p_i)} \tag{2.3}$$

où p_i représente la probabilité d'occurrence de la $i^{ème}$ faute, n est le nombre de fautes potentielles et m est le nombre de fautes détectées parmi les n.

2.7.1.2 Cas des fautes paramétriques

Supposons que nous avons n performances et m critères de test. Soit $A = (A_1, A_2, \ldots, A_n)$ l'ensemble des spécifications et $B = (B_1, B_2, \ldots, B_m)$ les limites de test. Les métriques de test sont calculées théoriquement comme suit :

$$Y = \int_A f_S(s)\, ds \tag{2.4}$$

$$Y_T = \int_B f_T(t)\, dt \tag{2.5}$$

$$Y_C = \frac{\int_A \int_B f_{ST}(s,t)\, ds\, dt}{Y} \tag{2.6}$$

$$D = 1 - \frac{\int_A \int_B f_{ST}(s,t)\, ds\, dt}{Y_T} \tag{2.7}$$

où $f_S(s) = f_S(s_1, s_2, \ldots, s_n)$ est la densité de probabilités conjointe des performances, $f_T(t) = f_T(t_1, t_2, \ldots, t_m)$ est la densité de probabilités conjointe des critères de test et $f_{ST}(s,t) = f_{ST}(s_1, s_2, \ldots, s_n, t_1, t_2, \ldots, t_m)$ est la densité de probabilités conjointe des performances et des critères de test.

Les métriques de test seront calculées à partir des données obtenues en utilisant la simulation de type Monte Carlo du circuit sous test. Puisque le nombre d'itérations ne peut pas être très élevé, au plus 1000 simulations pour un circuit complexe, il n'est pas possible de calculer directement ces métriques de test avec une précision de l'ordre du ppm. Ainsi, il est nécessaire d'utiliser une technique statistique pour ce calcul. Dans la section suivante, on suppose que les performances du circuit suivent une distribution multinormale [17]. Ceci va permettre d'obtenir les densités de probabilité décrites précédemment.

2.7.2 Calcul des métriques de test paramétriques sous l'hypothèse gaussienne

En supposant que la densité de probabilité conjointe des performances et des critères de test est multinormale, les données obtenues par la simulation de type Monte Carlo du circuit sont utilisées pour calculer la matrice de variance-covariance qui sera employée pour calculer la fonction de densité de probabilité.

Soit le vecteur $X = (X_1, X_2, \ldots, X_p)^T$ composé de p variables aléatoires, où X_j, $j = 1, 2, \ldots, p$, est une variable aléatoire de dimension 1, la covariance de X_i et X_j est une mesure de dépendance entre ces variables aléatoires et elle est définie par :

$$\nu_{X_iX_j} = Cov(X_i, X_j) = E(X_iX_j) - E(X_i)E(X_j) \tag{2.8}$$

où $E(.)$ dénote l'espérance mathématique. Si X_i et X_j sont indépendantes, la covariance $\nu_{X_iX_j}$ est nécessairement égale à zéro mais l'inverse n'est pas vraie. La covariance d'une variable aléatoire X_i avec elle-même est la variance :

$$\nu_{X_iX_i} = Cov(X_i, X_i) = \nu_{X_i} \tag{2.9}$$

La corrélation entre deux variables X_i et X_j est définie à partir de la covariance comme :

$$\rho_{X_iX_j} = \frac{\nu_{X_iX_j}}{\sigma_{X_i}\sigma_{X_j}} \tag{2.10}$$

où l'écart-type est défini par $\sigma_{X_i} = \sqrt{\nu_{X_i}}$.

L'avantage de la corrélation est qu'elle est indépendante de l'échelle de mesure, c'est à dire, un changement de l'échelle de mesure des variables ne change pas la valeur de la corrélation. Par conséquent, la corrélation est plus utile comme mesure de dépendance entre deux variables aléatoires que la covariance. La corrélation est en valeur absolue toujours inférieure à 1 et égale à zéro si les variables aléatoires X_i et X_j sont indépendantes.

Une évaluation empirique de ces quantités exige un certain nombre d'observations. Supposons que $\{x_i\}_{i=1}^n$ est un ensemble de n observations d'un vecteur X de variables aléatoires dans $\mathbb{R}^p$. Chaque observation x_i a p dimensions : $x_i = (x_{i1}, x_{i2}, \ldots, x_{ip})$, et elle correspond à une valeur observée du vecteur $X \in \mathbb{R}^p$. La covariance de deux variables aléatoires est alors estimée par :

$$V_{X_iX_j} = \frac{1}{n-1}(\sum_{k=1}^{n} x_{ik}x_{jk} - n\bar{x_i}\bar{x_j}) \tag{2.11}$$

et la variance d'une variable aléatoire est estimée par :

$$V_{X_i} = \frac{1}{n-1}(\sum_{k=1}^{n} x_{ik}^2 - n\bar{x_i}^2) \tag{2.12}$$

La corrélation de deux variables aléatoires et alors donnée par :

$$r_{X_iX_j} = \frac{V_{X_iX_j}}{s_{X_i}s_{X_j}} \tag{2.13}$$

avec $s_{X_i} = \sqrt{V_{X_i}}$

Les covariances théoriques entre toutes les variables aléatoires peuvent être mises sous forme matricielle (matrice de variance-covariance) :

$$\Sigma = \begin{pmatrix} \nu_{X_1} & \dots & \nu_{X_1X_p} \\ \vdots & \ddots & \vdots \\ \nu_{X_1X_p} & \dots & \nu_{X_p} \end{pmatrix} \tag{2.14}$$

L'estimation empirique de la matrice de variance-covariance est donnée par :

$$S = \begin{pmatrix} V_{X_1} & \dots & V_{X_1X_p} \\ \vdots & \ddots & \vdots \\ V_{X_1X_p} & \dots & V_{X_p} \end{pmatrix} \tag{2.15}$$

Soit X une variable aléatoire a p dimensions d'espérance mathématique $\mu = (\mu_{i1}, \mu_{i2}, \dots, \mu_{ip})^T$ et de matrice variance-covariance Σ. Si X est de distribution multinormale, alors la densité de probabilités de X est donnée par :

$$f(x) = \frac{1}{\sqrt{det(2\pi\Sigma)}} \exp\left[-\frac{(x-\mu)^T\Sigma^{-1}(x-\mu)}{2} \right] \tag{2.16}$$

La probabilité de chaque sous-ensemble $A \subset \mathbb{R}^p$ est donnée par la formule :

$$P(A) = \frac{1}{\sqrt{det(2\pi\Sigma)}} \int_{A_1} \dots \int_{A_p} \exp\left[-\frac{(x-\mu)^T\Sigma^{-1}(x-\mu)}{2} \right] dx_1\, dx_2 \dots dx_p \tag{2.17}$$

Ainsi, en utilisant l'hypothèse multinormale, il est possible de dériver les fonctions densités de probabilités qui doivent être intégrées en considérant les limites des variables aléatoires afin de calculer les métriques de test.

2.7.3 Estimation des métriques de test

Il est clair que pour un nombre réduit de performances et de critères de test, l'équation (2.17) de la section précédente est facile à évaluer. Toutefois, dans les cas pratiques où le nombre de performances et de critères de test du circuit sous test est élevé, cette quantité est difficile à calculer directement (intégrale multiple difficile à évaluer), d'où le recours à la simulation.

Les métriques de test, définies précédemment, peuvent être directement estimées en utilisant les estimateurs suivants :

$$\hat{Y} = \frac{\text{Nombre des circuits fonctionnels}}{N} \tag{2.18}$$

$$\hat{Y_T} = \frac{\text{Nombre de circuits passants le test}}{N} \tag{2.19}$$

$$\hat{Y_L} = \frac{\text{Nombre de circuits fonctionnels échouant le test}}{\text{Nombre de circuits fonctionnels}} \tag{2.20}$$

$$\hat{D} = \frac{\text{Nombre de circuits défaillants passant le test}}{\text{Nombre de circuits passants le test}} \tag{2.21}$$

où N est le nombre de circuits générés.

Par exemple, supposons l'application d'une méthode de test, dépendant uniquement de deux critères de test, sur un circuit sous test. Les données de cet exemple sont illustrées sur la Figure 2.5, où le nombre de circuits simulé est de 15 avec :

- 10 circuits fonctionnels ;
- 5 circuits défaillants.

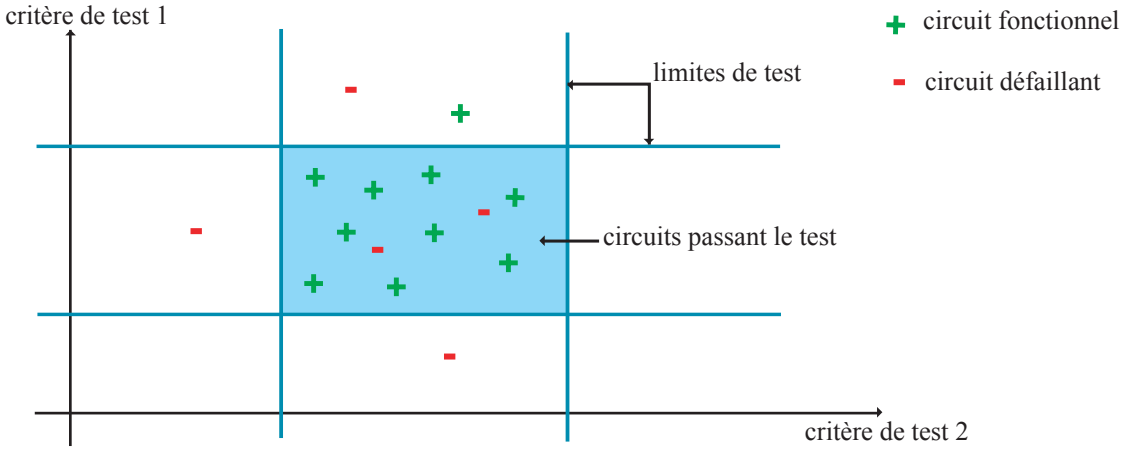


Figure 2.5 – Exemple des résultats du test d'un circuit.

L'estimation des métriques de test, suivant les formules précédentes (2.18 - 2.21) est :

$$\hat{Y} = \frac{10}{15} = 0.66$$

$$\hat{Y_T} = \frac{11}{15} = 0.73$$

$$\hat{Y_L} = \frac{1}{10} = 0.1$$

$$\hat{D} = \frac{2}{11} = 0.18$$

2.8 Conclusion

Dans ce chapitre, nous avons présenté le domaine du test des circuits mixtes et principalement des circuits analogiques. Dans la première section de ce chapitre, nous avons introduit quelques définitions des termes importants pour faciliter la lecture du document. Ensuite nous avons détaillé les notions de faute et de test.

Pour les circuits analogiques, nous avons distingué deux catégories de fautes : les fautes paramétriques et les fautes catastrophiques. Pour réduire le nombre de fautes à simuler, des méthodes d'analyse de fautes permettent d'extraire les fautes les plus probables d'un circuit à partir de sa vue physique «*layout*». Plusieurs travaux ont été publiés dans le domaine de la modélisation de fautes, mais contrairement aux circuits numériques, il n'existe pas de modèles de fautes standard pour les circuits analogiques.

Pour la simulation de fautes, l'un des problèmes majeurs est la prise en compte des tolérances dues aux variations du processus de fabrication. Il existe plusieurs travaux qui ont été publiés mais qui ne considèrent que certains types de circuits comme les circuits linéaires ou certains types d'analyse comme l'analyse DC. De plus, pour les circuits analogiques, les vecteurs de test dépendent de la nature du circuit à tester.

Actuellement la tendance est à l'utilisation des techniques de conception en vue du test DFT. L'ajout de structures de test pour les circuits analogiques est plus complexe que pour les circuits numériques. En effet, les circuits analogiques sont beaucoup plus sensibles que les circuits numériques. Malgré cette complexité, ces techniques de DFT commencent à être développées et notamment le test intégré BIST, ce qui nécessite de développer des techniques de test efficaces pour pouvoir évaluer avec précision les améliorations apportées par ces techniques de DFT.

Chapitre 3

État de l'art des méthodes d'ordonnancement et de réduction de tests

3.1 Introduction

Le test des circuits analogiques est un problème difficile, principalement dû à la nature non déterministe des signaux et à l'accessibilité limitée aux nœuds internes du circuit sous test. Les méthodes de test des circuits analogiques sont traditionnellement classifiées en deux catégories : structurelles (test des défauts) et fonctionnelles (test des spécifications). Pour le test structurel, un modèle de fautes, habituellement au niveau circuit, est adopté et des vecteurs de test (stimuli) sont appliqués au circuit sous test. Ensuite, on exploite la différence entre la structure d'un circuit fonctionnel et celle d'un circuit défectueux pour déterminer le bon fonctionnement ou non d'un circuit. Cependant, il y n'a pas de modèles de fautes universels pour tous les circuits analogiques. Pour le test fonctionnel, il s'agit de mesurer les performances du circuit sous test, telle que le gain, la fréquence de coupure [1], la vitesse de montée ou vitesse de descente [2], ..., et déterminer que le circuit est fonctionnel quand les mesures sont comprises entre les spécifications des performances correspondantes. Cette approche de test est facile à appliquer, cependant, elle manque d'information sur la structure du circuit sous test et elle est très coûteuse en temps et en équipement de test.

Les circuits analogiques doivent généralement satisfaire touts les spécifications du cahier des charges pour garantir leur bon fonctionnement. Le test de toutes les performances engendre une augmentation du temps de test ainsi que son coût. Il est alors plus efficace de tester un sous-ensemble de performances, tout en s'assurant que le nombre de circuits défaillants qui passent le test (taux de défauts) soit minimal.

1. Cut-off frequency
2. Slew rate

Un but important dans le test est la réduction du temps de test. Dans le domaine numérique, ceci correspond à déterminer l'ensemble des vecteurs de test qui maximise la couverture de fautes. En raison de la difficulté à définir des modèles de fautes, comme mentionné précédemment, il y a peu de recherche dans le domaine analogique. Huss *et al* [18] ont étudié le problème en ordonnant les tests de telle sorte que les circuits défectueux soient détectés tôt dans le flow des tests pour réduire le temps moyen de test. Cette approche est efficace pour réduire le temps de test d'un circuit défectueux mais n'a aucune incidence sur un circuit fonctionnel. Milor *et al* [19] propose aussi un algorithme pour trouver un ordre des tests pour augmenter l'efficacité du test fonctionnel. Cet algorithme élimine les tests non critiques en se basant sur une estimation du rendement. Cependant cette estimation n'est pas précise et est très gourmande en temps de calcul surtout quand les corrélations entre les tests sont considérées. Souders *et al* [20] présentent une approche pour choisir un ensemble minimal de vecteurs de test pour déterminer le comportement d'un convertisseur analogique-numérique. Cette méthode réduit le temps de test mais elle a besoin de points d'accès supplémentaires aux nœuds internes du circuit.

Il est difficile d'établir une classification générale de toutes les méthodes de test, visant à réduire le nombre de tests à effectuer. Dans ce chapitre, nous répartissons les différentes méthodes suivant les techniques utilisées dans l'ordonnancement ou bien la réduction des tests. Les principales classes sont :

1. Méthodes basées sur la corrélation.
2. Méthodes basées sur l'estimation des métriques de test.
3. Méthodes basées sur l'identification des paramètres.
4. Méthodes basées sur la classification.

3.2 Méthodes basées sur la corrélation

3.2.1 Définition

La relation entre deux variables aléatoires peut être quantifiée par leur covariance. Cependant, à l'image de la moyenne et de la variance, la covariance est un moment qui possède une dimension ce qui la rend plus difficile à interpréter. C'est pourquoi on utilise plus généralement le coefficient de corrélation, indicateur sans dimension, défini par

$$\rho(X,Y) = \frac{cov(X,Y)}{\sigma_X \sigma_Y} \tag{3.1}$$

Le coefficient de corrélation mesure la relation linéaire entre deux variables aléatoires X et Y (i.e. de la forme $Y = aX + b$). On a les propriétés suivantes :

- $\forall X, Y : \rho(X,Y) \in [-1,1]$.
- Si X et Y sont indépendantes, alors $\rho(X,Y) = 0$ (la réciproque n'est pas vraie en général).

– $\forall X, Y\ \forall a_1, a_2, b_1, b_2 \in \mathbb{R}\ (a_1 a_2 \neq 0) : \rho(a_1 X + b_1,\ a_2 Y + b_2) = sign(a_1 a_2)\rho(X, Y)$.
– S'il existe une relation linéaire entre X et Y alors $\rho(X, Y) = \pm 1$.

Brockman *et al* [21] exploite la corrélation entre les tests pour réduire leur nombre. Une fois que le sous-ensemble de tests à mesurer est identifié, un modèle de régression est construit pour calculer la valeur des tests éliminés. Une autre méthode décrite par Han *et al* [22] exploite la corrélation pour réduire l'ensemble des tests à effectuer sur un circuit. Ces travaux seront détaillés dans les deux sections suivantes (3.2.2, 3.2.3).

3.2.2 Méthode de l'analyse de la redondance

Dans [22], les auteurs proposent une méthode dynamique pour l'élimination des tests, applicable au test des fautes paramétriques des circuits analogiques. Un concept de redondance de test est introduit. Il exploite les corrélations entre les tests. A partir des données issues de la simulation ou bien de la production, les corrélations entre les tests sont calculées. De cette analyse, un intervalle de redondance est construit pour chaque test. Si la mesure d'un test est située dans l'intervalle de redondance d'un autre test, ce dernier test est éliminé de l'ensemble des tests. Cette approche réduit le coût du test dans le cas où les tests sont fortement corrélés mais n'est pas applicable dans le cas où les tests ne sont pas corrélés.

Les données issues de la simulation ou de la production sont utilisées pour analyser la redondance de chaque paire de tests. Pour chaque paire de tests sélectionnée, une régression polynomiale est construite en se basant sur la méthode des moindres carrés. Supposons que les erreurs de la régression polynomiale sont normalement distribuées avec une variance σ^2. On peut définir un intervalle de confiance pour la valeur $\hat{p}$ du test estimé (variable dépendante). La valeur réelle du test est comprise dans l'intervalle $[\hat{p}-n\sigma, \hat{p}+n\sigma]$ avec une certaine probabilité, où $\hat{p}$ représente la valeur estimée du test, σ l'écart-type de l'erreur estimée, et $\pm n\sigma$ est l'intervalle de confiance. Par exemple, $n = 3.5$ implique un intervalle de confiance à 99.9%.

Dans cette approche, l'intervalle de redondance d'un test est défini comme la région du test indépendant (variable indépendante), où une décision succès/échec peut être prise pour le test dépendant. L'intervalle de redondance dépend de la corrélation entre les deux tests, la distribution de chacun des tests, et des limites de test.

Dans l'exemple de la Figure 3.1, le calcul de la redondance est montré dans le cas d'un circuit avec deux tests t_1 et t_2. La Figure 3.1(a) montre la courbe de régression ainsi que les bornes de l'intervalle de confiance. Supposons que le bon fonctionnement du circuit exige que le test t_2 soit supérieur ou égal à y ($[y, +\infty]$). De la Figure 3.1(a), on peut dire qu'un circuit sous test va satisfaire le test t_2 (variable dépendante) si sa valeur au test t_1 (variable indépendante) est inférieur à x. Alors, l'intervalle de redondance du test t_2 dans le domaine du test t_1 est $[-\infty, x]$. Cet intervalle de redondance est illustré par la région colorée de la Figure 3.1(b). Ainsi, dans l'intervalle de redondance la décision

de succès/échec du test t_2 ne nécessite pas une mesure réelle du test t_2, elle est déduite directement de la valeur de test t_1. Hors de l'intervalle de redondance, la décision de succès/échec du test t_2 exige une mesure réelle du test t_2.

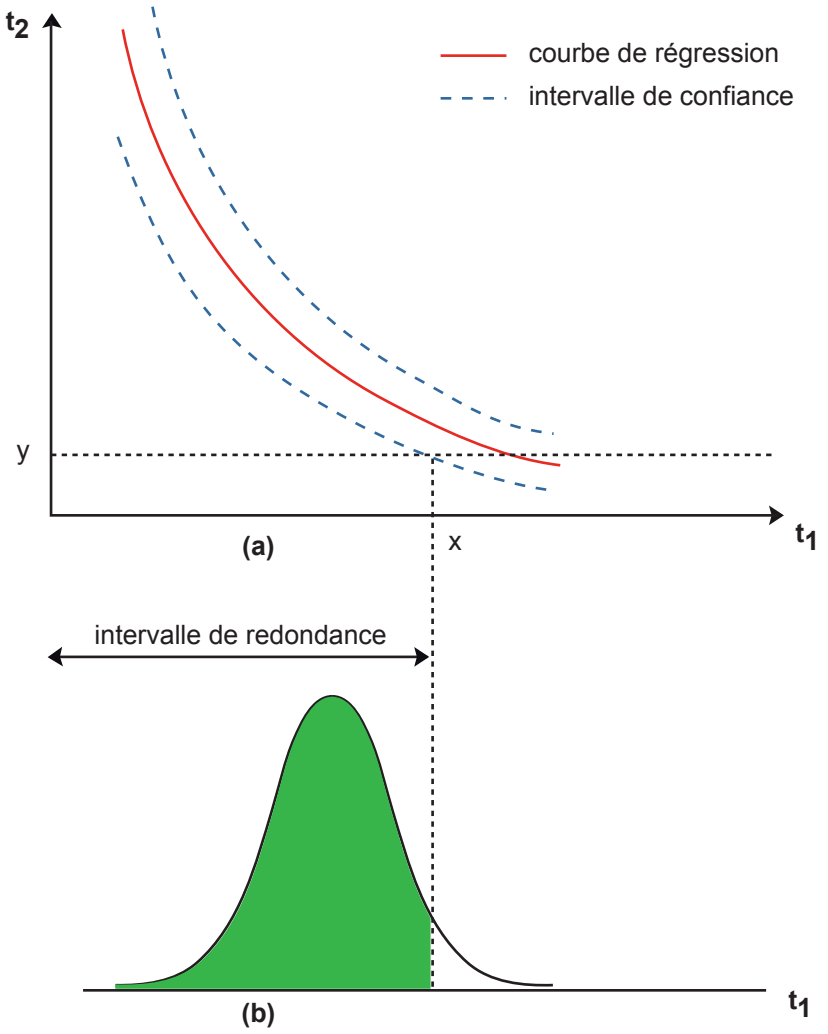


Figure 3.1 – Exemple de construction d'un intervalle de redondance : (a) ajustement polynomial avec intervalle de confiance et (b) redondance du test t_2 dans le domaine du test t_1.

La matrice des redondances T est obtenue à partir de la définition de l'intervalle de redondance et de la matrice des corrélations. L'élément t_{ij} de T est déterminé comme suit :

1. Si la paire de tests i et j a un coefficient de corrélation $\rho(i, j)$ inférieur à un seuil, alors $t_{ij} = 0$. Ceci implique que le test j ne peut pas être estimé à partir du test i. Un seuil est nécessaire car dans la réalité, une corrélation parfaite ($\rho = \pm 1$) est impossible.

2. Sinon, t_{ij} est la probabilité que la valeur du test i soit comprise dans l'intervalle de redondance du test j. Alors, t_{ij} représente la probabilité de succès/échec du test j sachant la valeur du test i.

$$T = \begin{bmatrix} t_{11} & t_{12} & \dots & t_{1n} \\ t_{21} & \ddots & & \vdots \\ \vdots & & & \\ t_{n1} & & \dots & t_{nn} \end{bmatrix} \tag{3.2}$$

La matrice des redondances aide à identifier les tests susceptibles d'être éliminés. Toutefois, le coût du test ne peut pas être déterminé uniquement sur la base de la matrice des redondances T. Supposons un ensemble de n tests avec des temps de test $W_i, i = 1, \dots, n$. Alors, en générale, le temps moyen de test t_{avg} peut être calculé par :

$$T_{avg} = \sum_{i=1}^{n} Wi \left[\prod_{j=1}^{i-1} Y_j \right] \tag{3.3}$$

où Y_j est le rendement du test j.

La combinaison de l'équation (3.3) avec la notion de redondance de test permet de récrire t_{avg} sous la forme :

$$T_{avg} = \sum_{i=1}^{n} Wi \left[\prod_{j=1}^{i-1} Y_j \right] P(N_i | G_1 \cap \dots \cap G_{i-1}) \tag{3.4}$$

où le terme G_k est l'événement stipulant que le circuit sous test passe le $k^{ième}$ test et N_k l'événement définissant la non élimination du $k^{ième}$ test de l'ensemble des tests par l'analyse de la redondance. Ainsi, la probabilité $P(N_i|G_1 \cap \dots \cap G_{i-1})$ représente la fonction de vraisemblance de la mesure réelle du $i^{ième}$ test sachant que tous les précédents tests $(1, \dots, (i-1))$ sont satisfaits. Si le $i^{ième}$ test est redondant (déterminé à partir la matrice des redondances T), $P(N_i|G_1 \cap \dots \cap G_{i-1})$ tend vers 0. Par contre, si le $i^{ième}$ test n'est pas corrélé avec les tests $1, \dots, (i-1)$, alors $P(N_i|G_1 \cap \dots \cap G_{i-1})$ tend vers 1 et l'équation (3.4) devient identique à l'équation (3.3). Toutes les méthodes d'ordonnancement de test, comme dans [18] (voir la section 3.6.1) et [23], peuvent être utilisées pour réduire le temps de test, en utilisant l'équation (3.4) comme fonction coût.

3.2.3 Méthode de prédiction de tests par régression polynomiale

Brockman *et al* [21] ont introduit une méthode qui permet la prédiction d'un sous-ensemble de tests [3] basée sur un modèle statistique des variations paramétriques du processus de fabrication. En évaluant la distribution de probabilité conjointe de l'ensemble des performances du circuit, la méthode exploite la corrélation entre les performances pour

3. Predictive subset testing

réduire le nombre de performances à tester explicitement. Une fois le sous-ensemble de performances identifié, des modèles de régression sont construits pour prédire les performances non testées. A partir des intervalles de confiance, les limites de test sont assignées aux performances testées. Le résultat est un sous-ensemble de tests qui réduit le coût et la complexité du test.

Pour construire le sous-ensemble de tests, on teste explicitement une performance X et on utilise cette mesure pour prédire la valeur de la performance Y. Après la simulation électrique de n instances du circuit sous test, on extrait de chaque circuit la mesure des deux performances dans les paires $(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)$. En supposant une régression polynomiale entre X et Y, on obtient :

$$Y_i = \beta_0 + \beta_1 X_i + \beta_2 X_i^2 + \ldots + \beta_{p-1} X_i^{p-1} + \epsilon_i \; (i = 1, \ldots, n) \tag{3.5}$$

où $p-1$ est l'ordre du modèle et $\beta_0, \beta_1, \ldots, \beta_{p-1}$ sont les paramètres de régression qui peuvent être estimés par la méthode des moindres carrés. Des observation $(X_i, Y_i), i = 1, \ldots, n$, on définit les vecteurs et la matrice suivante :

$$\mathbf{Y} = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix} \quad \mathbf{X} = \begin{bmatrix} 1 & X_1 & X_1^2 & \ldots & X_1^{p-1} \\ 1 & X_2 & X_2^2 & \ldots & X_2^{p-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & X_n & X_n^2 & \ldots & X_n^{p-1} \end{bmatrix}$$

$$\beta = \begin{bmatrix} \beta_0 & \beta_1 & \beta_2 & \ldots & \beta_n \end{bmatrix}^T$$

Pour une valeur donnée $X = x$, on définit le vecteur

$$\mathbf{x} = \begin{bmatrix} 1 & x & x^2 & \ldots & x^{p-1} \end{bmatrix}^T \tag{3.6}$$

et on suppose que la variable aléatoire Y peut être exprimée sous la forme :

$$Y = \beta^T \mathbf{x} + \epsilon \tag{3.7}$$

où la variable aléatoire ϵ est l'erreur des observations par rapport à la droite de régression. Pour simplifier, on suppose que ϵ suit une loi normale avec une moyenne nulle est une variance σ^2. Comme x est donnée dans l'équation (3.6), la quantité $\beta^T \mathbf{x}$ est une constante. Alors pour chaque observation $(X_i, Y_i), i = 1, \ldots, n$, nous faisons les hypothèses (Gauss-Markov) suivantes pour Y_i :

- Indépendance : chaque observation Y_i est indépendante statistiquement.
- Variance constante : chaque Y_i a la même variance σ^2.
- Normalité : Pour $X_i = x_i, Y_i$ suit une loi normale avec une moyenne $\beta^T \mathbf{x_i}$ et une variance σ^2.

A fin d'établir l'intervalle de confiance d'une nouvelle occurrence de Y, on va construire un test de Student (t-test). L'estimation des moindres carrés de β est :

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}. \tag{3.8}$$

Si la moyenne et la variance de la distribution de la performance prédite sont inconnues, leurs valeurs sont reliées à la performance testée explicitement à travers la courbe de régression. Pour une valeur donnée $X = x_h$, une estimation sans biais de la moyenne de la performance prédite est :

$$\hat{Y}_h = \hat{\beta}^T\mathbf{x_h}. \tag{3.9}$$

En assumant les trois hypothèses de Gauss-Markov, la statistique

$$t^*_{h(new)} = \frac{\hat{Y}_h - Y_{new}}{s(\hat{Y}_h - Y_{new})}. \tag{3.10}$$

est distribuée suivant une loi de Student avec $n - p$ degré de liberté, où Y_{new} est la valeur de la nouvelle observation de Y et

$$s(\hat{Y}_h - Y_{new}) = MSE(1 + \mathbf{X}_h^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}_h) \tag{3.11}$$

avec MSE[4] l'erreur quadratique moyenne donnée par :

$$MSE = \frac{\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2}{n - p}. \tag{3.12}$$

Pour un seuil de risque α, l'intervalle de confiance $1 - \alpha$ pour prédire une nouvelle occurrence de Y est :

$$\hat{Y}_h \pm t(1 - \alpha/2, n - p)\, s(\hat{Y}_h - Y_{new}). \tag{3.13}$$

Une métrique simple pour vérifier l'ajustement du modèle de régression est le coefficient de détermination $r^2, 0 \leq r^2 \leq 1$, qui mesure la fraction de la variation de Y expliquée par la régression sur X

$$r^2 = \frac{\sum_{i=1}^{n}(Y_i - \overline{Y})^2 - \sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2}{\sum_{i=1}^{n}(Y_i - \overline{Y})^2} \tag{3.14}$$

Cette méthode utilise les corrélations entre les performances pour former un modèle de régression. Le test explicite d'une performance est utilisé pour prédire la valeur de la performance non testé.

Les deux méthodes de cette classe s'accordent dans l'utilisation de la corrélation pour éliminer les tests redondants. Toutefois, elles divergent dans l'utilisation des tests éliminés.

4. Mean Square Error

Tandis que Han *et al* [22] éliminent définitivement les tests redondants, Brockman *et al* [21] construisent des modèles de régression entre les performances testés pour prédire la valeur des tests éliminés et tiennent en compte de leurs valeurs dans le test du circuit.

3.3 Méthodes basées sur l'estimation des métriques de test

3.3.1 Définition

Les métriques de test ont été définies dans le chapitre précédent (section 2.7.1). Plusieurs méthodes utilisent des métriques de test pour ordonner les tests ou bien réduire leur nombre. Dans cette thèse, on utilisera la métrique du taux de défauts pour ordonner les tests en vue de leur élimination [24]. D'autres méthodes ([25], [19]) utilisent le rendement (proportion des circuits fonctionnels) et l'optimisation de la couverture de fautes pour réduire l'ensemble des tests. Une version améliorée de cette approche de réduction de l'ensemble des tests est présentée par Acar *et al* [26] où le coût du test est tenu en compte. Bounceur *et al* [27] fixent les limites de test comme compromis entre différentes métriques de test. Une fois les limites de test fixées, le sous-ensemble de tests ayant la plus grande couverture de fautes catastrophiques est choisi pour réduire le coût du test. La méthode de l'estimation du rendement et optimisation du temps de test [19] ainsi que la méthode de minimisation des erreurs de test par modélisation statistique [27] seront détaillées dans les deux sections suivantes.

3.3.2 Méthode de l'estimation du rendement et optimisation du temps de test

Milor *et al* [19] proposent une nouvelle approche pour ordonner les tests d'un circuit. Son principe consiste à éliminer les tests non critiques en se basant sur une estimation du rendement.

Supposons un ensemble de n tests à exécuter sur s circuits. La complexité requise pour trouver l'ensemble réduit des tests est $O(sn^2)$. Le temps moyen de test de production varie non seulement avec le nombre de tests qui doivent être réalisés, mais également varie selon l'ordre des tests, car le test se termine au premier test échoué [5]. Par conséquent, le temps moyen de test dépend non seulement du temps pour accomplir tous les tests, mais également de la probabilité qu'un test particulier sera réalisé sur un circuit. Alors, il est préférable de réaliser d'abord les tests courts qui sont le plus susceptibles d'échouer.

5. Stop on fail

Supposons un ensemble de n tests ayant les temps de test $W_i (i = 1, \ldots, n)$, alors le problème est de minimiser le temps de test moyen :

$$Temps\ moyen = \sum_{i=1}^{n} W_i \left[\sum_{j=1}^{i-1} Y_j \right] \tag{3.15}$$

où Y_j est le rendement du $j^{ième}$ test.

Initialement l'ensemble complet des tests est appliqué sur un échantillon de circuits fabriqués. Ces données peuvent donc être employées pour trouver le meilleur ordre des tests, minimisant le temps moyen du test de production, en essayant toutes les permutations possibles de l'ensemble des tests incluant seulement les premiers k tests nécessaires pour réaliser la couverture de fautes désirée. Elles peuvent aussi servir à calculer le temps moyen de test pour chaque ensemble de tests, et déterminer l'ensemble de tests ayant le plus petit temps moyen de test. Mais si le circuit a n performances, $n!$ permutations devraient être essayées. Pour chaque permutation, on doit calculer le temps moyen de test, qui est la somme du produit des temps de test et des probabilités qu'un circuit passe un sous-ensemble de tests. En particulier, en présence des variations du processus, les données de succès/échec doivent être stockées pour tous les tests. Alors, trouver les probabilités de passer tous les sous-ensembles possibles de tests est équivalent au calcul :

$$\prod_{j \in J} Y_j, \forall J \subset I \tag{3.16}$$

où I est l'ensemble de tous les tests. Si I a n tests, alors on a 2^n sous-ensembles possibles J. Par conséquent, 2^n probabilités de passer un sous-ensemble de tests doivent être calculées.

Soit une fonction aléatoire z_i pour un sous-ensemble J, telle que :

$$z = \begin{cases} 1 & \text{tous les tests dans } J \text{ sont satisfaits} \\ 0 & \text{au moins un test dans } J \text{ est violé} \end{cases}$$

Alors, le rendement du sous-ensemble J est approché par :

$$\prod_{j \in J} Y_j = \frac{1}{n} \sum_{i=1}^{n} z_i \tag{3.17}$$

L'approche de minimisation du temps de test considérée dans [18] est plus performante. Dans ce travail, l'algorithme de Dijkstra est appliqué pour choisir et ordonner les tests. La solution optimale est obtenue. Dans [18], le problème de sélection des tests est formulé comme un problème de recherche du *plus cours chemin* dans un graphe orienté (Figure 3.2). Chaque nœud du graphe représente un état possible, qui représente dans le cas du problème posé ici, un ensemble de tests. Chaque arc représente un chemin entre un état et un autre, et le poids associé à l'arc représente le coût moyen d'exécution du prochain test.

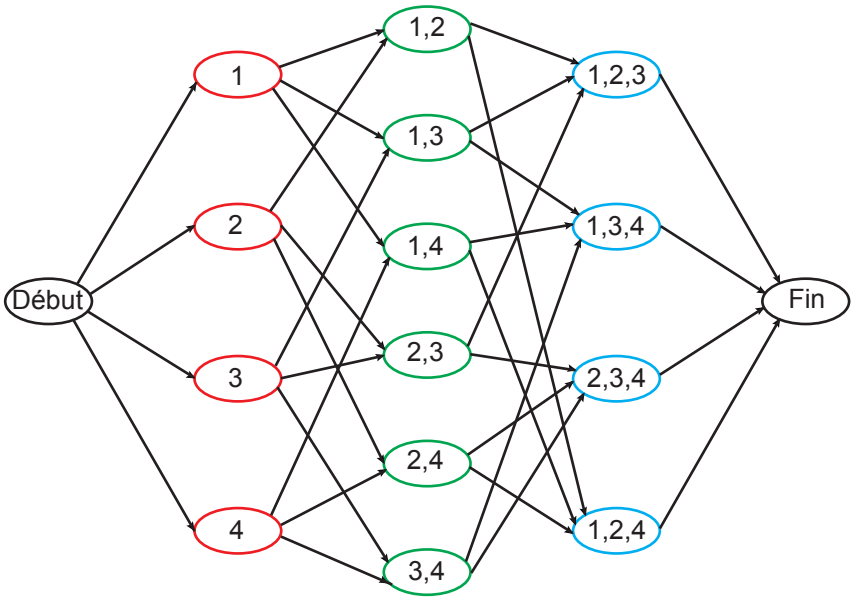


Figure 3.2 – Graphe orienté.

L'application de l'algorithme de Dijkstra n'est pas réalisable sur des circuits dont le nombre de tests dépasse 20. Le problème avec l'algorithme de Dijkstra est qu'il ne tient pas compte de la couverture de fautes, qui est critique dans l'ordonnancement des tests. Généralement, il est recommandé de tester, en premier, les tests avec la plus grande couverture de fautes pour minimiser le coût du test. Les algorithmes qui sont présentés dans [19] sont partiellement basés sur ces observations. Les tests éliminés en premier sont ceux qui ont une couverture de fautes très basse dans l'ensemble des tests, ensuite les tests restants sont ordonnés. Deux approches pour l'ordonnancement des tests sont considérées. Une approche commence avec un ensemble de tests ordonnés de telle sorte que les tests exécutés en premier aient la plus grande couverture de fautes. Ensuite, on essaye un ensemble de permutations qui sont susceptibles de réduire le temps de test. La deuxième approche est l'algorithme de recherche A^* [6], qui ordonne les tests de façon optimale en utilisant un graphe orienté, dont chaque nœud est associé à un temps de test estimé. Comme certains tests sont éliminés de l'ensemble des tests avant de les ordonner, les deux algorithmes sont des heuristiques. Ces deux heuristiques permettent de traiter des circuits avec un plus grand nombre de tests, et en un temps de calcul inférieur à l'application directe de l'algorithme de Dijkstra sur l'ensemble des tests.

6. A^* algorithm

3.3.3 Méthode de minimisation des erreurs de test par modélisation statistique

Dans [27], les auteurs ont pour objectif l'estimation des métriques de test analogique, en présence de multiple déviations paramétriques et en présence de fautes. Un modèle statistique est utilisé pour fixer les limites de test comme compromis entre les métriques de test, calculées pendant la phase de conception du circuit. Ce modèle est obtenu à partir de simulation Monte Carlo, en supposant que les déviations des paramètres et des performances du circuit suivent une loi multinormale. Après avoir fixé les limites de test en considérant les déviations du processus, les métriques de test sont calculées en présence de fautes catastrophiques et de fautes paramétriques simples.

En supposant que la loi conjointe des performances et des critères de test est multinormale. Les données de la simulation Monte Carlo sont utilisées pour estimer les paramètres de la loi multinormale : le vecteur des moyennes m et la matrice de variance-covariance V.

$$f(x, m, V) = \frac{1}{\sqrt{|V|(2\pi)^n}} \exp\left[-\frac{1}{2}(x-m)^T V^{-1}(x-m)\right] \tag{3.18}$$

Une fois le modèle multinormal validé, ce dernier est simulé pour générer un grand échantillon de données (1 million d'instances). Cet échantillon est utilisé pour estimer les métriques de test avec une précision de l'ordre du ppm, et particulièrement le taux de défauts $\hat{D}$ et la perte de rendement $\hat{Y_L}$ (section 2.7.1).

$$\hat{Y_L} = \frac{\text{Nombre de circuits fonctionnel échouant au test}}{\text{Nombre de circuit fonctionnel}} \tag{3.19}$$

$$\hat{D} = \frac{\text{Nombre de circuits défaillants passant le test}}{\text{Nombre de circuit passant le test}} \tag{3.20}$$

Fixer les limites de test est un compromis entre le coût du test et sa qualité. Dans cette étude, les limites de test sont fixées de telle manière à minimiser, simultanément, le taux de défaut et la perte de rendement. Les critères de test sont privilégiés car plus faciles à obtenir et moins coûteux que les performances. Une fois les limites de test des critères de test fixées, la couverture de fautes est calculée dans le cas des fautes paramétriques et catastrophiques. Pour chacun des deux types de fautes, l'ensemble minimal composé de performances et de critères de test maximisant la couverture de fautes est construit.

Une comparaison des deux méthodes exposées dans cette section montre une utilisation différente de l'estimation des métriques de test pour réduire et ordonner les tests d'un circuit. Dans [19], le rendement est estimé à partir de données de production alors que dans [27], une modélisation statistique est utilisée pour une meilleure estimation du taux de défauts et de la perte de rendement. Dans les deux méthodes, la couverture de fautes

catastrophiques est utilisée pour réduire le nombre de tests. Toutefois, l'ordre des tests est très important dans [19] tandis qu'il n'est pas considéré dans [27].

3.4 Méthodes basées sur l'identification des paramètres

3.4.1 Définition

Le test orienté modèle MBT [7] consiste à générer des tests depuis un modèle représentant le circuit à tester. Le modèle décrit le comportement attendu du système. Sur la base de ce modèle, des cas de test [8] sont générés automatiquement. Un cas de test est un ensemble de conditions ou de variables, sur lequel un testeur va déterminer si le circuit fonctionne correctement ou non. À partir de ces cas de test, une comparaison est possible entre le comportement réel du circuit et le comportement attendu (décrit dans le modèle).

Le problème du test d'un circuit peut être converti en un problème d'identification des paramètres d'un modèle comportemental [28]. Une fois les paramètres du modèle identifiés à partir de mesures particulières sur un circuit sous test, les performances peuvent être estimées avec le modèle au lieu de faire toutes les mesures physiques. Le principe du test est illustré sur la Figure 3.3.

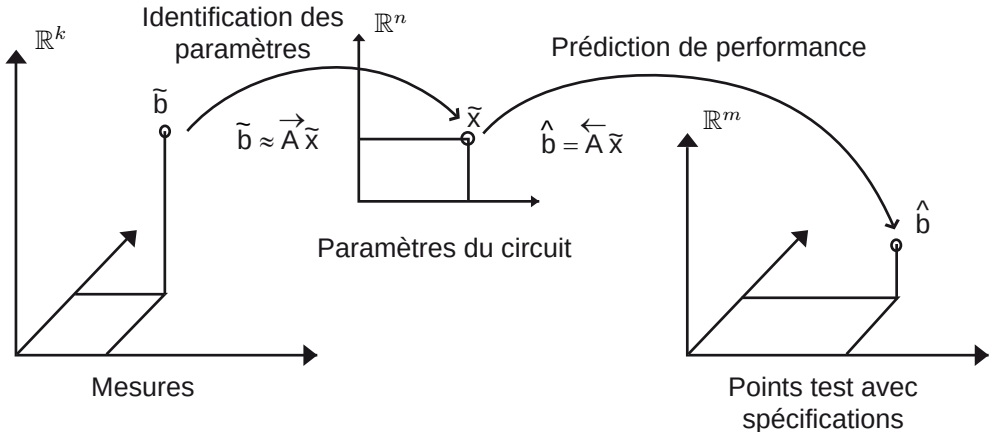


Figure 3.3 – Principe du test avec l'identification des paramètres.

Chaque circuit sous test peut être associé à un ensemble de n paramètres qui déterminent les performances obtenues aux m points de mesure [9]. Avec un modèle complet qui tient bien en compte les mécanismes d'erreur et une fois les paramètres du circuit connus, on peut prédire la valeur de toutes les performances et pour toutes les entrées possibles

7. Model Based Testing
8. Test case
9. Test points

du circuit. Alors, le test d'un circuit ne requiert pas les m points de mesure, mais uniquement un petit nombre k de mesures clef qui permettent d'identifier l'ensemble des $n \leq k$ paramètres du modèle d'un circuit sous test particulier.

L'exemple d'un convertisseur analogique numérique (CAN) de 13 bits est décrit dans [20], où un modèle linéaire avec $n = 18$ paramètres est utilisé. L'identification des paramètres du modèle est basée sur la mesure d'un sous-ensemble bien choisi de $k = 64$ codes convertis. Ces mesures sont notées dans la Figure 3.3 par $\tilde{b}$ et le système d'équations linéaires :

$$\tilde{b} \approx \overrightarrow{A}\tilde{x} \tag{3.21}$$

est résolu pour $\tilde{x}$ avec la méthode des moindres carrées. En calculant $\hat{b} = \overleftarrow{A}\tilde{x}$, les performances prédites du circuit pour la totalité des m codes peuvent être utilisées pour vérifier si les spécifications du circuit sont respectées.

Sans la validation orientée modèle, les performances du circuit doivent être mesurées pour l'ensemble des m codes. En comparaison, la mesure de l'effort pour l'identification des paramètres du modèle peut être significativement plus petite. Dans le cas de l'exemple [20], l'effort est plus petit d'un facteur de $\frac{m}{k} = \frac{8192}{64} = 128$.

En résumé, l'approche MBT permet de réduire l'effort de mesure des ressources de calcul. Toutefois, la construction du modèle ainsi que l'évaluation des performances demandent un effort majeur durant la phase de caractérisation du produit.

3.4.2 La méthode LEMMA

Dans [29], les auteurs utilisent l'algorithme LEMMA [10] avec des modifications pour améliorer son exécution de manière significative. Cette technique est un outil efficace pour tester des circuits analogiques et mixtes qui réduit au minimum le nombre de mesures exigées pour caractériser la fonction de transfert d'un circuit en déterminant un nombre restreint de paramètres d'un modèle linéaire d'erreur et puis en prévoyant l'erreur générale de la réponse. La méthode LEMMA est fondée sur l'hypothèse que la réponse d'un circuit est déterminée principalement par un nombre restreint de variables et dépend linéairement des déviations de ces variables par rapport à leurs valeurs nominales. Au lieu de mesurer la réponse entière du circuit, un nombre limité de mesures est fait sur des points soigneusement choisis et ceux-ci sont employés pour déterminer les coefficients du modèle. Ce dernier est employé pour prévoir la réponse entière du circuit.

L'erreur de la réponse $\mathbf{e}(m \text{ x } 1)$ du circuit sous test doit être écrite comme une combinaison linéaire des n vecteurs colonnes $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_n$:

$$\mathbf{e} = \mathbf{E}\mathbf{x} \tag{3.22}$$

10. Linear Error-Mechanism Model Algorithm

où les colonnes de $\mathbf{E}(m \text{ x } n)$ sont les vecteurs de la base $\mathbf{e}_i$ et $\mathbf{x}$ un n-vecteur avec des poids réels. Au lieu de mesurer les m composantes de l'erreur de la réponse totale, on détermine à sa place le vecteur des n poids de $\mathbf{x}$ à partir des p ($\geq n$) mesures pour prédire $\mathbf{e}$.

La première étape consiste à choisir la dimension n du modèle. Ce problème a été largement étudié dans la littérature d'identification des systèmes. La méthode SVD [11] [30] est employée pour déterminer le rang d'une matrice comportant des vecteurs de sensibilité et des signatures mesurées d'erreur ; alors le principe de parcimonie [31] est suivi dans la construction de $\mathbf{E}$.

Le système d'équations (3.22) est déterminé car nous avons m équations avec n inconnues et $m > n$. En plus, la présence du bruit de mesure dans $\mathbf{E}$ et $\mathbf{e}$ rend les équations incompatibles. Pour contrer ceci, nous mesurons p composantes de $\mathbf{e}$, où $n \leq p << m$, et estimons le vecteur des poids $\mathbf{x}$ par un ajustement par la méthode des moindres carrés $\mathbf{x_{LS}}$.

Le problème de détermination des points à mesurer dans l'erreur de réponse s'appelle la sélection de point test [12]. Si nous supposons que $t_1, t_2, \ldots, t_n$ indiquent les points de mesure choisis, et $\mathbf{e_R}(p \text{ x } 1)$ le vecteur des erreurs de réponse associées, alors nous obtenons $\mathbf{x_{LS}}(n \text{ x } 1)$ par la minimisation de

$$\|\mathbf{E_R x_{LS}} - \mathbf{e_R}\|$$

où $\mathbf{E_R}(p \text{ x } n)$ est le modèle réduit formé par les lignes $t_1, t_2, \ldots, t_p$ de $\mathbf{E}$.

En conclusion, nous estimons l'erreur de réponse par

$$\mathbf{e} \approx \mathbf{E x_{LS}}$$

Un choix optimal de n, $\{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_n\}$, p et $\{t_1, t_2, \ldots, t_p\}$ produit une estimation proche de la véritable erreur de réponse $\mathbf{e}$ et peut être employée pour vérifier la conformité par rapport aux spécifications. Plusieurs techniques sont employées dans [29] pour un choix optimal des différents paramètres clefs de la méthode LEMMA.

Une fois le modèle développé hors ligne, un algorithme de test est appliqué en ligne pour distinguer les circuits fonctionnels et défectueux. Les principales étapes de cette algorithme sont résumées dans l'algorithme 3.1.

11. Singular Value Decomposition
12. Test point selection

Entrées : Le modèle **E**, les points de mesure $t_1, t_2, \ldots, t_p$, la matrice pseudo-inverse associée $\mathbf{E_R^+}$
Mesurer les erreurs de la réponse aux p points de mesure et construire le vecteur des erreurs réduit $\mathbf{e_R} = (e_{t_1}, e_{t_2}, \ldots, e_{t_p})^T$.
Calculer le vecteur des coefficients $\mathbf{X_{LS}} = \mathbf{E_R^+ e_R}$.
Calculer $\mathbf{e} = \mathbf{EX_{LS}}$
si chaque élément de **e** est dans les limites d'erreur spécifiés **alors**
le circuit est fonctionnel
sinon
le circuit est défectueux
fin si
Sorties : Décision succès/échec du circuit sous test.

ALGORITHME 3.1 – Algorithme de test de la méthode LEMMA.

3.4.3 Méthode de sélection des points de mesure

La méthode de sélection des points de mesure présentée dans [32] part du postulat qu'un modèle précis du circuit sous test est déjà défini, comme dans [33]. Une fois que le modèle est déterminé, des opérations algébriques simples lui seront appliquées selon les étapes suivantes :

1. Sélectionner un ensemble optimal de points de mesure qui minimise le coût du test et maximise sa fiabilité.
2. Estimer les paramètres du modèle à partir des mesures sur les points de mesure sélectionnés.
3. Prédire la réponse du circuit dans tous les points de mesure comme base pour accepter ou rejeter un circuit.
4. Calculer la précision des paramètres estimés et la prédiction de la réponse, basées sur une erreur de mesure aléatoire.
5. Tester la validité du modèle.

Soit le modèle du circuit sous la forme suivante :

$$y^k = A\, x^k \tag{3.23}$$

où

$y^k = [y_1^k, y_2^k, \ldots, y_m^k]^T$ est le vecteur de la réponse du $k^{ième}$ circuit pour tous les m points de mesure.

les colonnes de la matrice A sont les n vecteurs du modèle a_j

$a_j = [a_{j1}, a_{j2}, \ldots, a_{jm}]^T$ est le $j^{ième}$ vecteur du modèle

$x^k = [x_1^k, x_2^k, \ldots, x_n^k]$ est le vecteur des n paramètres d'erreur du $k^{ième}$ circuit.

La réponse du $k^{ième}$ circuit au $i^{ième}$ point test est donnée par :

$$y_i^k = \sum_{j=1}^{n} a_{ij} x_j^k = a_{i1} x_1^k + a_{i2} x_2^k + \ldots + a_{in} x_n^k \tag{3.24}$$

s'il y a n paramètres d'erreur x_j, alors il suffit d'avoir n équations (3.24) pour trouver les valeurs des x_j. Alors un système réduit des équations est suffisant :

$$\tilde{y}^k = \tilde{A}\, x^k \tag{3.25}$$

$\tilde{A}$ et $\tilde{y}$ sont associés à l'ensemble réduit des n points de mesure.

L'ensemble réduit des n points de mesure utilisé dans l'équation (3.25) est obtenu par une décomposition QR [13] de l'équation (3.23).

L'équation (3.25) est résolue suivant l'ensemble réduit des mesures par :

$$\hat{x}^k = \tilde{A}^{-1}\, \tilde{y}^k \tag{3.26}$$

où $\hat{x}$ représente les variables estimées à partir des données de mesure.

Si le nombre de points de mesure sélectionnés est plus grand que le nombre minimum n, alors A n'est plus carrée, c'est-à-dire, il y a plus de lignes que de colonnes et l'équation (3.25) est résolue en utilisant une méthode des moindres carrés :

$$\hat{x}^k = (\tilde{A}^T\, \tilde{A})^{-1} \tilde{A}^T \tilde{y}^k \tag{3.27}$$

Dans ce cas, le résidu de la solution par les moindres carrés est calculé par :

$$\epsilon^k = \tilde{y}^k - \tilde{A}\, \tilde{x}^k \tag{3.28}$$

La moyenne quadratique (RMS [14]) de ϵ^k est une bonne mesure de la précision du modèle ; elle ne doit pas être sensiblement plus grande que le bruit de mesure.

Une bonne stratégie pour l'ajout de points de mesure supplémentaires est le calcul de la variance de prédiction à tous les points de mesure candidats sur la base des points de mesure déjà sélectionnés, puis de sélectionner le prochain point de mesure celui qui a la variance de prédiction la plus élevée. Le processus est répété jusqu'à ce que la variance de prédiction maximale soit réduite au niveau désiré. La variance de prédiction normalisée peut être calculée par :

$$\sigma_p^2/\sigma^2 = \text{diag}\,[A(\tilde{A}^T \tilde{A})^{-1} A^T] \tag{3.29}$$

où σ_p^2 est le $m-$vecteur de la variance de prédiction et σ^2 est la variance de chaque mesure, supposée constante.

13. Q : matrice orthogonale et R : matrice triangulaire supérieur.
14. Root Mean Square

Après avoir évalué les variables en utilisant l'équation (3.26) ou (3.27), la réponse est prédite en tous les points de mesure candidats y^k, en effectuant la multiplication matricielle de l'équation (3.23).

Une comparaison entre les deux méthodes de cette section montre que la méthode proposée dans [29] est plus élaborée. Par contre, la méthode exposée dans [32] est plus facile à mettre en œuvre et avec des logiciels libres (non commerciaux).

3.5 Méthodes basées sur la classification

3.5.1 Définition

Les méthodes basées sur la classification visent à discriminer entre les circuits fonctionnels et défectueux en utilisant des frontières de séparation déterminées dans l'espace des mesures (ou signatures). A partir d'un échantillon de circuits, la technique consiste à entraîner un module de classification (classificateur). Chaque élément de l'échantillon est classé comme fonctionnel ou défaillant suivant qu'il respecte ou non les spécifications des performances. Les frontières sont ensuite déterminées afin de séparer au mieux les éléments fonctionnels et défaillants. Toutefois, il peut y avoir des erreurs de classification (Figure 3.4) dues au mélange des éléments fonctionnels et défaillants. Ces erreurs sont aussi dues à l'incertitude sur l'estimation des signatures. L'objectif est donc de déterminer par apprentissage les frontières qui minimisent les erreurs de classification.

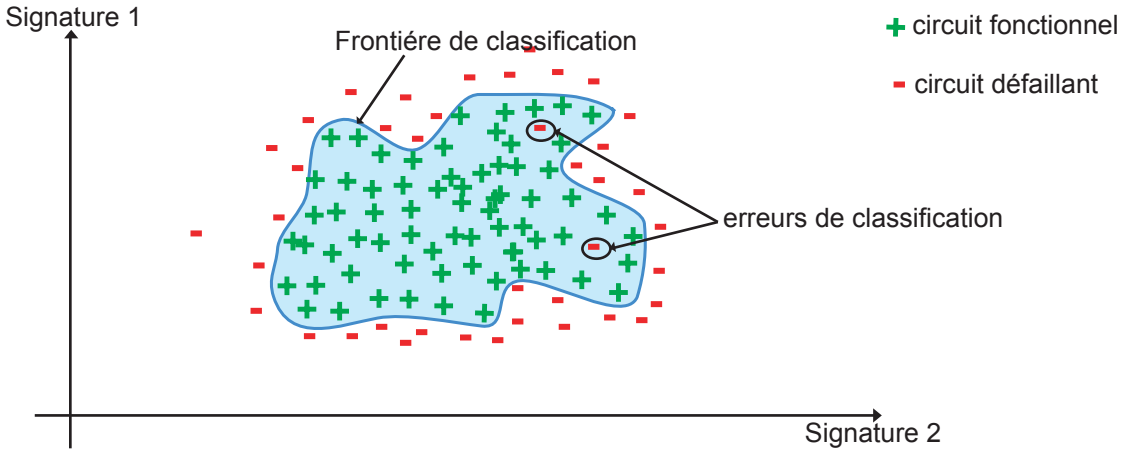


Figure 3.4 – Exemple de classification dans un espace à deux signatures.

Stratigopoulos *et al* [34] comparent l'utilisation de quatre différentes méthodes de classification pour éliminer les tests RF très coûteux en temps et en équipement de test. L'ensemble des tests restant (non RF) peut être complété par quelques tests RF pour améliorer l'efficacité et la fiabilité du test. Dans [35], les auteurs utilisent un algorithme

génétique multi-objectifs pour la réduction du nombre de tests RF. Cette approche permet de choisir un sous-ensemble de tests ainsi que le classificateur approprié. Biswas *et al* [36, 37] proposent d'éliminer les tests redondants en utilisant un algorithme d'apprentissage. Les principales étapes des deux méthodes de [36] et [35] seront détaillés dans les sections suivantes (3.5.2, 3.5.3).

3.5.2 La méthode ε-SVM

Dans [36], les auteurs proposent d'éliminer les tests superflus en utilisant la méthode de classification ε-SVM [15] [38], pour obtenir un test économique qui peut vérifier toutes les spécifications du circuit sous test, en identifiant et en éliminant les tests redondants en utilisant les méthodes d'apprentissage statistique. La méthode commence par l'ensemble complet des tests, donc sans aucune perte de rendement ni de taux de défauts. Les tests superflus sont éliminés jusqu'à ce que la prévision de l'erreur dépasse un seuil de tolérance défini par l'utilisateur. Durant l'élimination des tests, un modèle statistique est construit pour prédire le comportement du circuit (succès/échec) basé sur les tests restants. Puisque la quantité d'erreurs prévues à partir du modèle statistique défini est limitée, le procédé permet de contrôler la perte de rendement et le taux de défauts. En plus, quand le modèle statistique fait une erreur de prévision, la partie examinée est considérée appartenir à une région incertaine. Les circuits appartenant à cette région peuvent être réexaminés pour éviter une perte de rendement.

Soit un ensemble de points $\{(X_1, y_1), \ldots, (X_l, y_l)\}$, où $X_k = [X_k^1, \ldots, X_k^m]$ est une entrée à m-dimensions et y_k une sortie à une dimension. La méthode d'apprentissage statistique est un processus de construction d'un modèle pour la relation inconnue $y = f(X)$ utilisant l'ensemble des points donnés. Les données qui sont employées pour construire le modèle s'appellent les données d'apprentissage [16], alors que les données utilisées pour vérifier le modèle s'appellent les données de validation [17].

Dans ce travail, la méthode ε-SVM a été utilisé pour la construction du modèle d'apprentissage statistique. Le but dans ε-SVM est de construire un hyper-plan qui divise l'entrée X_k en deux classes $y_k = 1$ et $y_k = -1$. Ayant un ensemble de données d'apprentissage $\{(X_1, y_1), \ldots, (X_l, y_l)\}$ et une erreur limite prédéfinie ϵ, cette classification est réalisée en établissant une fonction $f(X)$ pour estimer y_k tel que : (1) pour chaque (X_k, y_K), $f(X_k)$ a tout au plus une déviation $E + \epsilon_k$ de y_k , où $\epsilon_k \geq 0$ est une erreur (2) $\sum_{k=1}^{l} \epsilon_k$ est réduit au minimum, et (3) le nombre de maximums et minimums locaux de $f(x)$ est minimisé. Ceci signifie que l'erreur d'estimation de y_k de (X_k, y_k) dans $\{(X_1, y_1), \ldots, (X_l, y_l)\}$ est inférieure à E. Cependant, il peut y avoir des (X_k, y_k) où $|y_k - f(X_k)| > \epsilon$, et l'erreur

15. ε Support Vector Machines
16. Training data
17. Test data

additionnelle est notée ϵ_k. La valeur $e_m = y_k - f(X_k)$ est l'erreur du modèle de $f(x)$ au point (X_k, y_k).

Comme le test est un problème de classification en succès/échec, ε-SVM est bien adapté à la réduction du nombre de tests. En plus, ε-SVM donne des résultats suffisamment précis en un temps raisonnable.

Le processus de réduction du nombre de tests commence par l'ensemble complet des tests, l'ensemble de données d'apprentissage et les spécifications de chaque test. En se basant sur les données, le succès ou l'échec de chaque dispositif est obtenu et ajouté aux données d'apprentissage. Après, le processus d'élimination commence avec l'ensemble des tests T, ce qui signifie que T_{red} est vide donc sans aucune perte de rendement ni de défauts initiale. Un test $t \in \{T - T_{red}\}$ est alors choisi pour une élimination potentielle et enlevé de l'ensemble de données d'apprentissage. On utilise ε-SVM sur les données d'apprentissage réduites pour établir un modèle de prédiction succès/échec pour l'ensemble des spécifications éliminées S_{red}. Le nombre total des dispositifs qui sont mal classés suivant S_{red} définit l'erreur de prédiction e_p. S'il est possible de construire un modèle d'ε-SVM pour avoir une erreur de prévision sur les données test doit être inférieurs à la tolérance définie par l'utilisateur e_T, le test choisi est considéré superflu et il est éliminé de manière permanente de l'ensemble des tests. Si, d'autre part, le modèle ne peut être créé avec suffisamment d'exactitude, le test est considéré nécessaire et ajouté de nouveau à l'ensemble des tests pour être testé. Ce processus est répété jusqu'à ce que chaque test $t \in T$ soit examiné. À la fin de ce processus, un ensemble réduit de tests et un modèle prévoyant le succès ou l'échec d'un circuit est obtenu. Les principales étapes de cette méthode sont détaillées dans la Figure 3.5.

3.5.3 Méthode de réduction des tests RF

La méthode de réduction des tests RF exposée dans [35], utilise un algorithme génétique multi-objectif pour explorer tous les sous-ensembles de tests et associe à chacun d'entre eux le meilleur classificateur. Le sous-ensemble le moins coûteux est choisi pour réduire le coût du test.

Soient N circuits avec $S = [s_1, \ldots, s_d]$ l'ensemble des tests. Pour chaque circuit, on enregistre $s_k, k = 1, \ldots, d$, et une étiquette de son succès ou échec aux tests. L'ensemble des N circuits est décomposé en un ensemble d'apprentissage et un ensemble de validation. Nous supposons que les étiquettes du succès/échec ne sont connues que pour les dispositifs de l'ensemble d'apprentissage ; les étiquettes de ces dispositifs sur l'ensemble de validation sont supposées être inconnues et elles ne sont utilisées que pour estimer l'erreur de test.

Un algorithme génétique est utilisé pour explorer les 2^d sous-ensembles de S. On assigne à chaque sous-ensemble visité $\acute{S} \subseteq S$ une évaluation basée sur deux critères : (a) son coût associé $C(\acute{S})$ et (b) l'erreur de test $\epsilon_r(\acute{S})$ lorsque de nouveaux dispositifs qui sortent de la production sont soumis uniquement à $\acute{S}$. Les circuits qui échouent à un ou plusieurs

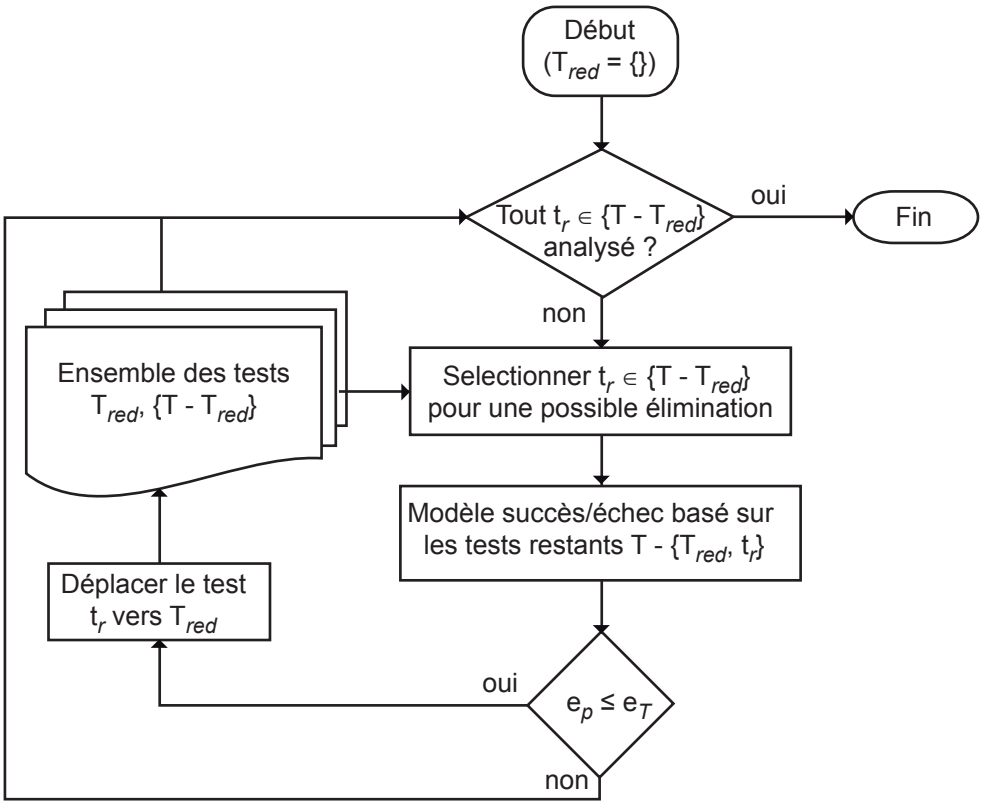


Figure 3.5 – Organigramme du processus d'élimination des tests.

tests dans $\acute{S}$ sont écartés. Les circuits qui passent tous les tests dans $\acute{S}$ sont présentés à un classificateur qui établit un modèle binaire de la forme $f : \acute{S} \rightarrow (\text{succès}, \text{échec})$. L'erreur $\epsilon_r(\acute{S})$ est définie comme le pourcentage des circuits dans l'ensemble de validation qui passent tous les tests dans $\acute{S}$ et qui sont mal classés par le modèle f.

On considère le cas général où l'ensemble des tests nécessite différents instruments de test et temps d'exécution. On partage l'ensemble des tests de S en M groupes. Si on note le nombre de tests dans un groupe i par n_i, alors $d = n_1 + n_2 + \ldots + n_M$ est le nombre total des tests. Soient respectivement $T(S)$ et $C(S)$ la référence du temps de test et le coût de test par seconde, quand tous les d tests sont considérés. Alors

$$C(S) = \sum_{i=1}^{M}(c_i C_{RF})(t_i T(S)) = C_{RF} T(S) \sum_{i=1}^{M} c_i t_i \tag{3.30}$$

où t_i est le temps de test relatif au groupe i par rapport à $T(S)$ et c_i est coût de test relatif au groupe i par rapport coût de test par seconde de l'équipement de test RF, noté ici par C_{RF}. Soit $x_{ik} = 1$ si le test k est présent dans le groupe i, et $x_{ik} = 0$ sinon. Soit aussi t_{ik} le temps de test du test k dans le groupe i. Le coût de test du sous-ensemble $\acute{S}$

est donné par :

$$C(\acute{S}) = \sum_{i=1}^{M} \left(c_i(1 - \overline{x_{i1}} \ldots \overline{x_{in_i}}) C_{RF} \sum_{k=1}^{n_i} t_{ik} x_{ik} \right) \tag{3.31}$$

avec "." l'opérateur "ET" logique.

Supposons que $t_{ik} = t_i T(S)/n_i$. L'équation précédente (3.31) devient :

$$C(\acute{S}) = C_{RF} T(S) \sum_{i=1}^{M} \left(\frac{c_i t_i}{n_i} \sum_{k=1}^{n_i} x_{ik} \right) \tag{3.32}$$

Le critère de performance du coût de test normalisé pour un sous-ensemble $\acute{S}$ est donné par :

$$C(\acute{S}) = \frac{C(\acute{S})}{C(S)} = \frac{(\sum_{i=1}^{M} (\frac{c_i t_i}{n_i} \sum_{k=1}^{n_i} x_{ik}}{\sum_{i=1}^{M} c_i t_i} \tag{3.33}$$

L'algorithme génétique explore le compromis $\epsilon - C$ avec pour objectif de converger vers la frontière de Pareto. La recherche continue jusqu'à ce qu'un objectif soit atteint ou bien qu'un nombre maximal d'itérations soit réalisé.

Deux classificateurs ont été utilisés dans cette méthode :

1. k-plus proches voisins [18] : cet algorithme constitue le plus simple classificateur non linéaire [39]. Étant donné un modèle $\acute{S}_u$, dont l'étiquette cible (succès ou échec) est inconnue, nous examinons ses k plus proches voisins dans un ensemble d'apprentissage (pour une valeur impaire de k), et attribuons à $\acute{S}_u$ l'étiquette ayant le plus grand nombre de représentants parmi les plus proches voisins. Cela équivaut à un vote à la majorité dans le voisinage de $\acute{S}_u$. Le choix de la valeur de k est dépendant des données. En règle générale, l'augmentation de k améliorent les résultats, jusqu'à un certain point au-delà duquel le résultat diminue. Un autre paramètre pertinent est la métrique de proximité. Dans ces expériences, on utilise la distance euclidienne pour déterminer la proximité du modèle.
2. Les réseaux de neurones (ONN [19]) : le deuxième classificateur employé dans cette méthode est ONN [40]. Il est entraîné pour former une hypersurface séparant les populations de dispositifs fonctionnels et défectueux dans le sous-espace défini par $\acute{S}$. Quand un nouveau modèle $\acute{S}_u$ est présenté à la formation des réseaux de neurones, il prend une décision de succès ou d'échec en fonction de l'empreinte de $\acute{S}_u$ par rapport à l'hyper-surface.

Les deux méthodes exposées dans cette section s'appuient sur la classification pour réduire le nombre de tests d'un circuit sous test. Dans [36], la méthode utilise uniquement le classificateur ε-SVM pour décider du succès ou de l'échec d'un circuit. Alors que la

18. k-Nearest Neighbors (K-NN)
19. Ontogenic Neural Network

méthode exposée dans [35] utilise un algorithme génétique pour combiner les avantages de deux classificateurs : les k-plus proche voisins et les réseaux de neurones.

3.6 Autres méthodes

Les différentes classes décrites précédemment permettent de bien organiser les méthodes d'ordonnancement des tests en quatre catégories bien distinctes. Toutefois, cette classification n'est pas exhaustive dû à la complexité du domaine du test analogique. Dans cette section, nous présentons d'autres méthodes spécifiques et non classables dans les quatre catégories précédentes.

Plusieurs travaux de recherche [41, 42, 43] qui allègent la difficulté de la génération de vecteurs de test, classification des fautes, amélioration de la qualité de test des circuits analogiques et mixtes ont été présentées, liant l'information des fautes structurelles et les performances du circuit sous test, ce qu'on appelle le test alternatif (voir la section 2.6.2). Dans [44], le test est optimisé en identifiant les tests qui détectent la majorité des défauts et en éliminant les tests non nécessaires et qui augmentent le temps de test. Une autre méthode proposée dans [18] ordonne les tests de telle sorte que les circuits défectueux soient détectés tôt dans la séquence de test pour réduire le coût du test. Cette dernière méthode est détaillée dans la section suivante.

3.6.1 La méthode des probabilités d'échec

Dans [18], Huss *et al* emploient la programmation dynamique pour ordonner les tests de telle sorte que des circuits défectueux soient détectés tôt. Une estimation précise des probabilités d'échec de chaque test ainsi que les probabilités communes d'échec de plusieurs tests sont exigées. Ces probabilités peuvent être estimées en utilisant des simulations ou à partir de circuits réels.

Le but d'ordonner les tests est de faire échouer le circuit aussi rapidement que possible s'il est défectueux. Les tests à placer en premier dans l'ordre des tests sont :

1. Tests qui ont une probabilité élevée d'échec.
2. Tests qui prennent peu de temps pour s'accomplir.
3. Tests qui sont indépendants des tests précédents.

Pour pouvoir ordonner des tests, une évaluation précise des probabilités d'échec d'un test ou d'échec de plusieurs tests est nécessaire. Ceci exige la collecte de données précises. Ces données peuvent être obtenues à partir de la simulation Monte Carlo ou bien à partir de données de production sur un testeur. A partir de ces données, une matrice m x n peut être construite, où m est le nombre de circuits examinés et n le nombre de tests. La

matrice A est :

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & & & \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \tag{3.34}$$

où $a_{i,k}$ est la valeur du paramètre k pour le circuit i. On suppose un vecteur des limites supérieures et inférieures, donné par les équations (3.35) et (3.36) respectivement.

$$T_U = (T_{O_1,U}\, T_{O_2,U} \dots T_{O_n,U}) \tag{3.35}$$

$$T_L = (T_{O_1,L}\, T_{O_2,L} \dots T_{O_n,L}) \tag{3.36}$$

Une expression du temps de test prévu est :

$$E(T) = \sum_{i=1}^{n} P(\{S_{O_{i-1}}\}) t_{O_i} \tag{3.37}$$

où $P(\{S_{O_{i-1}}\})$ est la probabilité que le circuit passe tout les tests ordonnés du premier jusqu'a la $j^{ième}$ position. $P(\{S_{O_0}\})$ est égale à 0 et t_{O_j} est le temps requis pour effectuer le test ordonné à la $j^{ième}$ position.

Il existe de nombreuses manières de prévoir les probabilités contenues dans (3.37). Ici, l'analyse de Monte Carlo sera employée. Cette méthode calcule les probabilités en comptant le nombre d'occurrences de l'évènement et en la divisant par le nombre d'occurrences possibles (le nombre de circuits examinés ou le nombre de simulations de Monte Carlo). La procédure de calcul des probabilités est résumée dans l'algorithme 3.2.

```
count ← 0
pour i Allant de 0 Jusqu'à m faire
  num ←= 0
  pour k Allant de 0 Jusqu'à j faire
    si  T_{O_k,L} ≤ a_{i,O(k)} ≤ T_{O_k,U} alors
      num ← num + 1
    fin si
  fin pour
  si num = j alors
    count ← count + 1
  fin si
fin pour
P({S_{O_j}}) = count/m
```

ALGORITHME 3.2 – Procédure de calcul des probabilités d'échec.

Le problème d'ordonnancement des tests implique de choisir des valeurs pour $O_i(i = 1, n)$. Une des manières évidentes de résoudre ce problème est d'évaluer toutes les possi-

bilités d'ordonner les tests, calculer la valeur de $E(T)$, et choisir l'ordre avec le minimum de $E(T)$. Cette méthode est pratique pour des circuits avec peu de tests. Cependant, elle n'est pas efficace pour des circuits dépassant six tests. Des techniques de programmation dynamique peuvent être employées afin de produire une solution optimale.

3.7 Conclusion

Afin de réduire les coûts et le temps de test, un ensemble de méthodes a été présenté dans ce chapitre. Une classification non exhaustive de ces méthodes a été introduite, permettant de représenter et de répartir la majorité des approches de ce domaine. Une analyse approfondie de toutes ces méthodes montre que :

1. Aucune des méthodes n'est généralisable à tous les circuits mais adapté uniquement à une catégorie particulière de circuits.
2. Toutes les méthodes ont besoin de circuits défectueux issus de la simulation ou bien de la production. Or, avec la robustesse des procédés de fabrication modernes, l'obtention de circuits défectueux est très coûteux en temps et en équipements.

Il est intéressant d'explorer des méthodes de réduction de performances plus générales et utilisant uniquement des circuits fonctionnels, plus faciles à obtenir en production ou bien par simulation.

Chapitre 4

Modélisation statistique

4.1 Introduction

La simulation Monte Carlo en micro-électronique est une méthode de simulation de circuits. Elle est très utilisée car elle permet de simuler un circuit tout en tenant compte des variations de ses paramètres internes. Celle-ci permet d'avoir un échantillon de données sur le comportement d'un circuit sous test. Cette technique est avantageuse par rapport à la mesure directe des données de production et peut être utilisée avant même la fabrication des premiers prototypes du circuit. Toutefois, avec des circuits de plus en plus complexes, la simulation Monte Carlo présente certains inconvénients. Le temps de simulation devient conséquent et requiert l'utilisation de moyens informatiques importants. En plus, la taille des échantillons obtenus par la simulation Monte Carlo est faible alors qu'on exige une précision de l'ordre du ppm, ce qui nécessite de disposer de plusieurs millions d'instances du circuit sous test.

Dans ce chapitre, nous allons utiliser la modélisation statistique basée sur un petit échantillon de données issues de la production ou bien de la simulation Monte Carlo. Une fois le modèle validé, le rééchantillonnage du modèle statistique permet la génération d'un grands échantillons d'instances du circuit sous test en peu de temps. Les modèles explorés comportent deux méthodes paramétriques : le modèle multinormal et les copules [45] ainsi que la méthode non paramétrique [46].

4.2 Modélisation multinormale

4.2.1 La loi normale

La loi de distribution, dite loi normale (ou loi normale gaussienne, loi de Laplace-Gauss), est la loi de variable aléatoire qui régit les variations de nombreux paramètres physiques. Elle est généralement applicable lorsque les dispersions de la variable sont dues à l'influence de nombreux paramètres indépendants les uns des autres et dont les

effets s'additionnent. Les variables qui ont de l'influence peuvent avoir des distributions normales, mais aussi quelconques.

4.2.1.1 Définition

On dit qu'une variable aléatoire X suit une loi normale, si la loi de distribution des probabilités de X est définie par une fonction de densité de probabilité de la forme :

$$f(x) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left[\frac{-(x-\mu)^2}{2\sigma^2}\right] \tag{4.1}$$

avec :

μ **:** l'espérance mathématique de la variable X;

σ **:** l'écart-type de de la variable X.

On note habituellement cela de la manière suivante : $X \sim \mathcal{N}(\mu, \sigma^2)$

On peut immédiatement voir que la loi normale est une loi symétrique autour de la valeur moyenne μ (si l'on remplace $x - \mu$ par $\mu - x$, $f(x)$ ne change pas). On peut aussi voir que la loi normale est complètement définie par son espérance μ et son écart-type σ.

Sur la Figure 4.1, la loi normale est illustrée pour différentes valeurs de μ et σ.

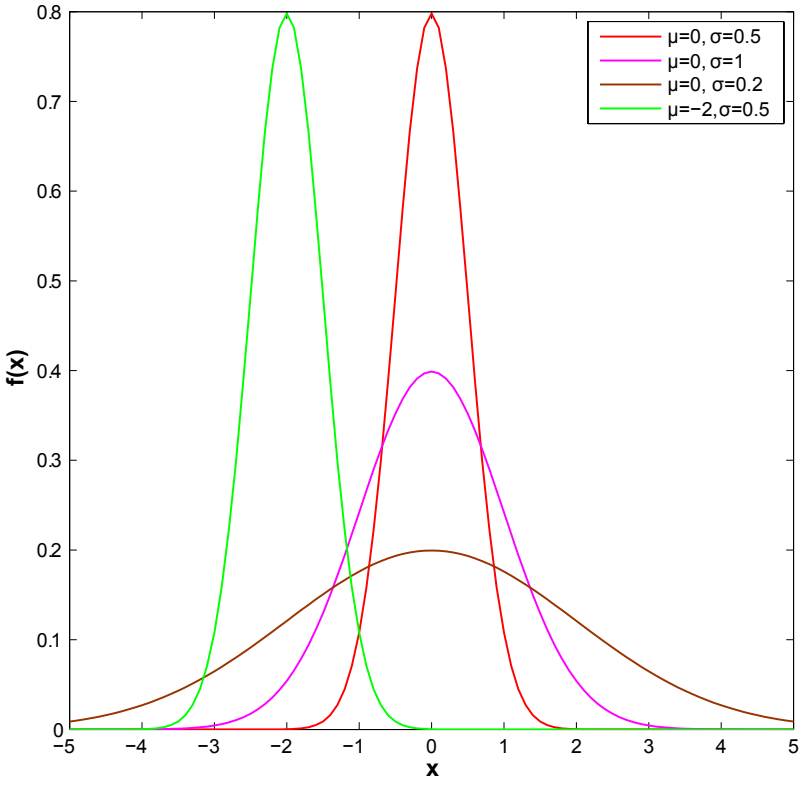


Figure 4.1 – Représentation graphique d'une loi normale pour différentes valeurs des paramètres μ et σ.

4.2.1.2 Loi normale centrée et réduite

Dans le cas particulier ou l'espérance ($\mu = 0$) et l'écart-type ($\sigma = 1$), la distribution de la loi normale devient :

$$z(x) = \frac{1}{\sqrt{2\pi}} \exp\left[\frac{-x^2}{2}\right] \tag{4.2}$$

et on dit qu'on a affaire à la loi normale centrée et réduite Z. On montre qu'on peut passer aisément d'une variable normale quelconque X de moyenne μ et d'écart-type σ, à une variable normale centrée et réduite ϵ par un changement de variable linéaire de la forme $\epsilon = \frac{X-\mu}{\sigma}$.

4.2.1.3 Caractéristiques de la loi normale

Pour une loi normale, les différentes mesures de tendance (moyenne, médiane et mode) sont toutes égales à μ. Bien que la fonction de densité ne soit jamais nulle, elle prend une valeur très petite en tout point x distant de plus de $3\,\sigma$ de la moyenne μ. Pour une distribution normale, la probabilité qu'une variable aléatoire X soit à l'intérieur de l'intervalle $[\mu - k\sigma, \mu + k\sigma]$ pour différentes valeurs de k est donnée dans le Tableau 4.1.

intervalle considéré	**probabilité**
$[\mu - 0.5\sigma, \mu + 0.5\sigma]$	38.29%
$[\mu - \sigma, \mu + \sigma]$	68.26%
$[\mu - 1.5\sigma, \mu + 1.5\sigma]$	86.63%
$[\mu - 2\sigma, \mu + 2\sigma]$	95.45%
$[\mu - 2.5\sigma, \mu + 2.5\sigma]$	98.75%
$[\mu - 3\sigma, \mu + 3\sigma]$	99.73%
$[\mu - 3.5\sigma, \mu + 3.5\sigma]$	99.95%
$[\mu - 4\sigma, \mu + 4\sigma]$	99.99%

Tableau 4.1 – Tableau des variations de la loi normale.

4.2.2 La loi multinormale

La loi multinormale ou loi normale sur $\mathbb{R}^n$ étend la loi normale à un vecteur aléatoire $X = (X_1, X_2, \ldots, X_n)$ à valeurs dans $\mathbb{R}^n$. L'équation (4.1) est généralisée à n-dimensions sous la forme :

$$f(x, m, V) = \frac{1}{\sqrt{|V|(2\pi)^n}} \exp\left[-\frac{1}{2}\,(x-m)^T\, V^{-1}\,(x-m)\right] \tag{4.3}$$

avec $|V|$ le déterminant de V, V^{-1} la matrice inverse de V et $(x-m)^T$ la transposée du vecteur $(x-m)$. Cette loi est habituellement notée $\mathcal{N}(m, V)$ par analogie avec la loi normale unidimensionnelle.

L'exemple d'une loi multinormale à 2 dimensions avec :

$$m = [0, 0] \text{ et } V = \begin{bmatrix} 0.25 & 0.3 \\ 0.3 & 1 \end{bmatrix}$$

est illustrée sur la Figure 4.2.

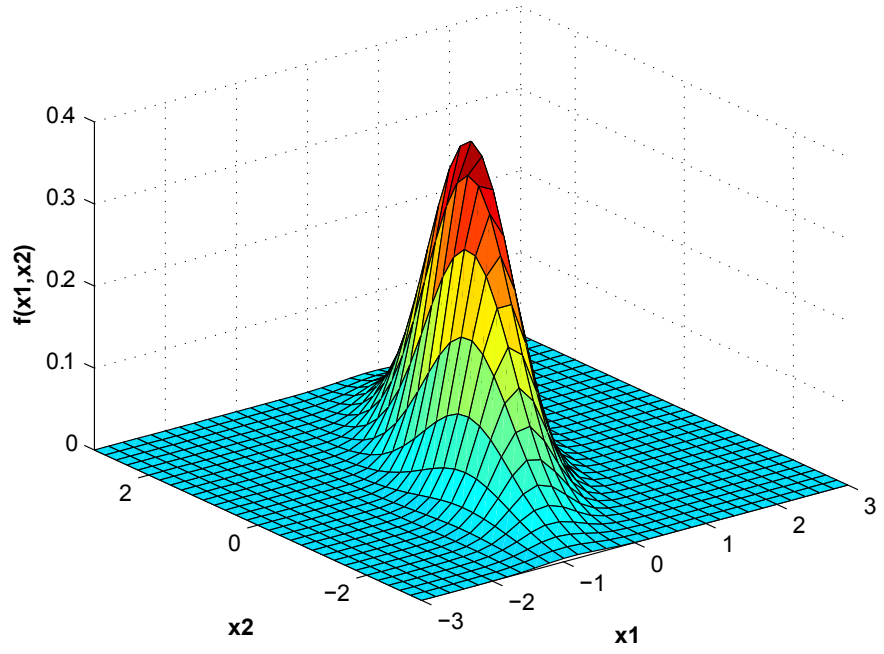


Figure 4.2 – Représentation graphique d'une loi multinormale à 2 dimensions.

La loi multinormale est caractérisée par deux paramètres : un vecteur m des espérances mathématiques, et une matrice V (carrée d'ordre n) de variance-covariance. Chaque élément m_i de m représente l'espérance de la variable aléatoire X_i. Chaque élément V_{ij} de V représente la covariance des variables aléatoires X_i, X_j et en particulier, chaque élément diagonal V_{ii} de V représente la variance σ_i^2 de la variable aléatoire X_i.

Comme la matrice V est une matrice de variance-covariance, elle est symétrique réelle, à valeurs propres positives ou nulles ; lorsque la loi multinormale est non dégénérée, la matrice V est à valeurs propres strictement positives : elle est définie positive ; dans ce cas, la loi multinormale admet une densité sur $\mathbb{R}^n$.

Pour simuler une loi multinormale non dégénérée de paramètres m et V, on utilise la méthode suivante :

1. Soit T un vecteur aléatoire à n composantes gaussiennes centrées réduites et indépendantes (la loi de T, multinormale, a pour moyenne le vecteur nul et pour matrice de variance-covariance la matrice identité).

2. Soit L la matrice résultante de la factorisation de Cholesky [1] de la matrice V.
3. Alors, le vecteur aléatoire $X = m + LT$ suit la loi multinormale de moyenne m et de matrice de variance-covariance V.

4.2.2.1 Estimation de l'espérance mathématique et de l'écart-type à partir d'un échantillon

Dans la pratique, on distingue l'espérance mathématique et l'écart-type théoriques d'une distribution de probabilités notés respectivement μ et σ de la moyenne et l'écart-type estimés ou observés sur un échantillon notés respectivement $\bar{x}$ et s. Il existe deux types d'estimation d'un paramètre inconnu : ponctuelle et par intervalle de confiance.

Estimation ponctuelle : l'estimation ponctuelle est choisie de façon à ce qu'elle converge vers la valeur du paramètre, qu'elle soit sans biais c'est à dire que son espérance mathématique soit égale à la valeur du paramètre et que sa variance soit minimum. Ainsi à partir d'un échantillon de taille n, les estimations de la moyenne μ et de l'écart-type σ sont données par :

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n} \tag{4.4}$$

$$s = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}} \tag{4.5}$$

Intervalle de confiance : le risque $\alpha \in [0, 1]$ d'une estimation correspond à la probabilité que l'estimation qui est faite d'un paramètre soit fausse. Donc, si on a une estimation à risque α, alors la probabilité que l'estimation soit bonne est égale à $(1 - \alpha)$.

L'estimation d'un paramètre par un intervalle de confiance au risque $\alpha \in [0, 1]$ consiste à trouver un intervalle qui a une probabilité de $(1 - \alpha)$ de contenir la vraie valeur du paramètre. Pour un échantillon de taille n, l'intervalle de confiance de la moyenne μ est défini par $\bar{x} \pm |\epsilon|\frac{s}{\sqrt{n}}$ pour les grands échantillons ($n > 30$) ou ϵ est obtenue à partir de la table de la loi normale centrée et réduite. Si l'échantillon est de petite taille ($n < 30$), par $\bar{x} \pm |t|\frac{s}{\sqrt{n}}$ ou t est la valeur maximale de la fonction t_{n-1} de Student d'ordre $(n - 1)$ correspondant au risque α choisi. Dans le Tableau 4.2, nous donnons les valeurs de ϵ et t de Student pour différentes valeurs du risque α. Le paramètre t de Student dépend du risque α et de la taille n de l'échantillon.

1. Consiste pour une matrice symétrique définie positive V, à déterminer une matrice triangulaire inférieure L telle que : $V = LL^T$.

α	**Probabilité correspondante**	$\|\epsilon\|$	$\|t\|(n=5)$	$\|t\|(n=10)$
0.20	80%	1.282	1.533	1.383
0.10	90%	1.645	2.132	1.833
0.05	95%	1.960	2.776	2.262
0.01	99%	2.576	4.604	3.250

Tableau 4.2 – Les valeurs des paramètres $|\epsilon|$ et $|t|$ pour différentes valeurs de α.

4.2.2.2 Validation du modèle multinormale

Il existe une riche littérature concernant le test de la loi multinormale [47, 48, 49, 50]. Malheureusement, il n'y a aucun test puissant et fiable [51]. Supposons que la population des circuits produite par la simulation de type Monte Carlo suit une distribution multinormale. Nous validons cette hypothèse en examinant la normalité des variables marginales dans l'espace indépendant correspondant à la projection des performances.

Supposons que la simulation Monte Carlo nous permet d'obtenir une matrice X, où pour chaque circuit de l'échantillon, on a la valeur de ses performances. Ces dernières suivent une loi multinormale de vecteur moyenne m et de matrice variance-covariance V.

Selon le théorème spectral [2], V peut être décomposée sous la forme $V = PDP^{-1}$, où

D : une matrice diagonale dont les coefficients sont des valeurs propres ;

P : une matrice dont les colonnes sont des vecteurs propres.

Pour le choix des vecteurs propres, nous pouvons faire en sorte que $P^{-1} = P^T$. Alors la décomposition de V pourra s'écrire sous la forme $V = PDP^T$.

Si X suit une loi multinormale $\mathcal{N}(m, V)$, alors $U = P^T(X - m)$ suit une loi normale $\mathcal{N}(0, D)$. Donc, le test de l'hypothèse gaussienne est équivalent au seul test de la normalité marginale du vecteur U (l'indépendance est assurée par le fait que D est diagonale) car :

$$E(U) = 0 \tag{4.6}$$

$$E(UU^T) = P^T E((X-m)(X-m)^T)P = P^T V P$$

$$= P^T P D P^T P = D \tag{4.7}$$

où $E(.)$ est l'espérance mathématique.

La vérification de la normalité des variables marginales est effectuée par la méthode de la droite de Henry [3]. Cette méthode permet de lire rapidement la moyenne et l'écart-type d'une telle distribution. Son principe est de tracer les quantiles de la performance à tester en fonction des quantiles théoriques d'une distribution normale standard (centrée réduite). Si la variable est gaussienne, les points doivent être alignés sur une droite définie par les données.

2. Soit A une matrice symétrique réelle, alors il existe une matrice P orthogonale et une matrice D diagonale dont tous les coefficients sont réels, telles que la matrice A est égale à PDP^{-1}.

3. Quantile-Quantile plot (Q-Q plot)

4.2.2.3 Exemple d'application

Nous allons vérifier l'hypothèse de normalité sur un amplificateur différentiel, conçu sous la technologie CMOS [4] $0.18\mu m$ de ST Microelectronics, avec la méthode de la droite de Henry. Ce circuit est formé de quatre blocs principaux : le circuit d'alimentation, le circuit de déclenchement, le circuit de contrôle du mode en commun et l'amplificateur lui même. La Figure 4.3 illustre ce circuit.

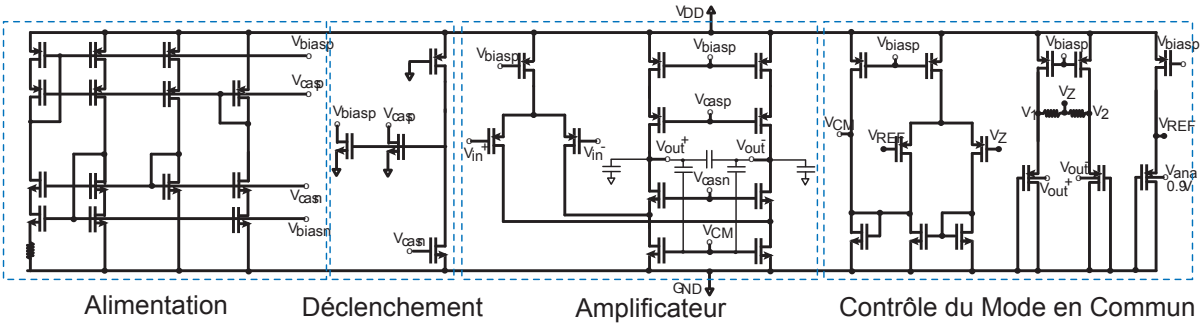


Figure 4.3 – Schéma du circuit sous test (amplificateur).

Une simulation de type Monte Carlo (1000 itérations) de cet amplificateur a été effectuée afin de trouver les différentes distributions gaussiennes ajustées à chacune de ses performances. L'ensemble complet des performances considérées est indiqué dans le Tableau 4.3, où a_1 et a_2 représentent les spécifications de chaque performance.

			Spécification	
Performance	μ	σ	a_1	a_2
1. A_D	76.60 dB	0.493 dB	74.49 dB	78.71 dB
2. GBW_D	330 MHz	18.14 MHz	252.36 MHz	407.64 MHz
3. *Phase Margin* (PM)	63.33°	0.45°	61.40°	65.26°
4. $CMRR$	-42.76 dB	1.02 dB	-47.13 dB	-38.39 dB
5. $PSRR$ (G_{ND})	-29.99 dB	3.65 dB	-45.61 dB	-14.37 dB
6. $PSRR$ (V_{DD})	-28.21 dB	3.75 dB	-44.26 dB	-12.16 dB
7. THD	66.19 dB	2.38 dB	56.00 dB	76.38 dB
8. *Current Consumption* (I_{DD})	2.48 mA	0.21 mA	1.58 mA	3.38 mA
9. *Intermodulation* ($Inter$)	67.57 dB	1.09 dB	62.90 dB	72.24 dB
10. $SR+$	73.14 V/μB	5.55 V/μB	49.38 V/μB	96.88 V/μB
11. $SR-$	73.14 V/μB	5.55 V/μB	49.38 V/μB	96.88 V/μB
12. *In Referred Noise* ($Noise$)	39.22 μV	0.5 μV	37.08 μV	41.36 μV

Tableau 4.3 – Paramètres gaussiens et spécifications des performances.

Les spécifications de l'amplificateur ne sont pas connues à priori, puisque l'application réelle du dispositif n'est pas considérée dans ce travail. Ainsi, pour avoir un rendement élevé de $Y = 99.99\%$ quand toutes les performances sont considérées, chaque spécification i est fixée à $\mu_i \pm k\sigma_i$, où k représente la marge de tolérance de chaque spécification dont la valeur est fixée à 4.3.

4. Complementary Metal Oxide Semi-conductor

L'analyse des corrélations entres les différentes performances montre que la performance $SR+$ a les mêmes valeurs que la performance $SR-$. Ainsi, cette dernière a été éliminée dans la matrice des corrélations. Dans la suite de l'étude, le nombre de performances considérées sera donc réduit à 11.

La Figure 4.4 montre l'application de la méthode de la droite de Henry sur toutes les variables marginales dans l'espace indépendant, issue de la projection des performances.

Nous remarquons sur chacune des figures, que le nuage de points est aligné, avec de légers écarts vers les limites des distributions. Ceci montre que les variables marginales correspondantes peuvent être approchées par des distributions gaussiennes. Ces résultats indiquent que l'hypothèse de multinormalité des performances n'est pas rejetée dans le cas de l'amplificateur opérationnel.

Des simulations de type Monte Carlo sont nécessaires pour obtenir les paramètres de la loi multinormale (vecteur des moyennes et matrice de variance-covariance) des performances du circuit sous test. La Figure 4.5 illustre la simulation de 1000 instances suivant la loi multinormale et 1000 instances de la simulation de type Monte Carlo, de 2 performances du circuit sous test.

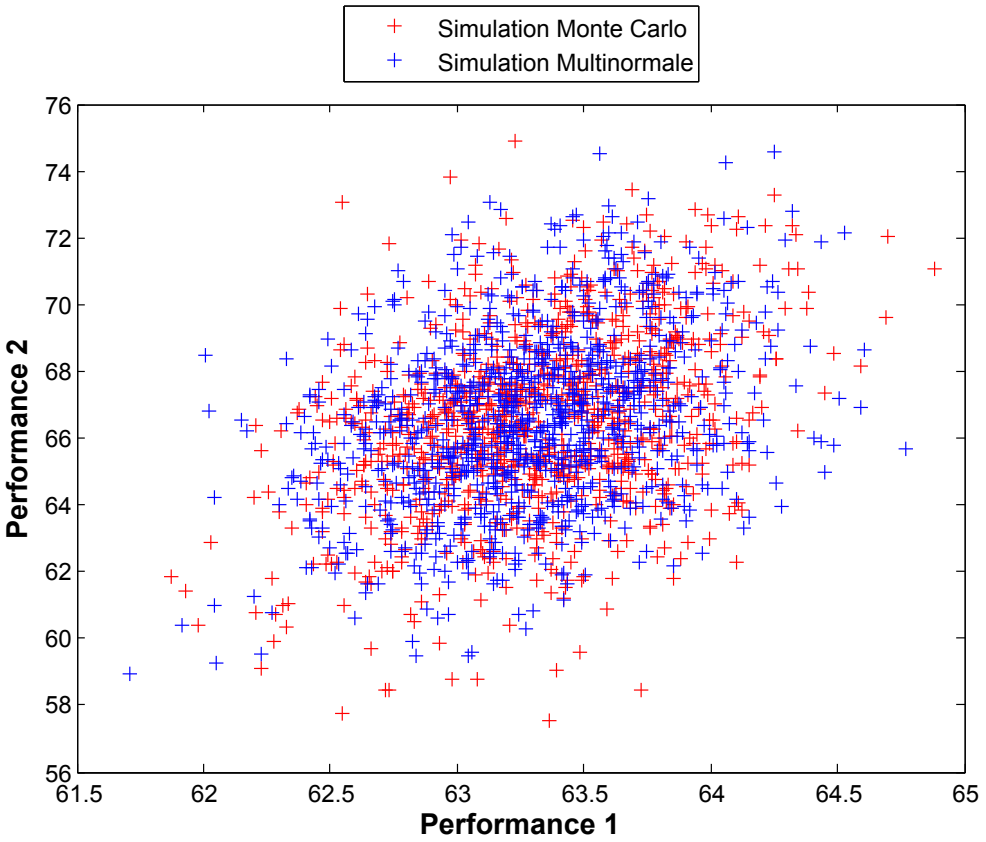
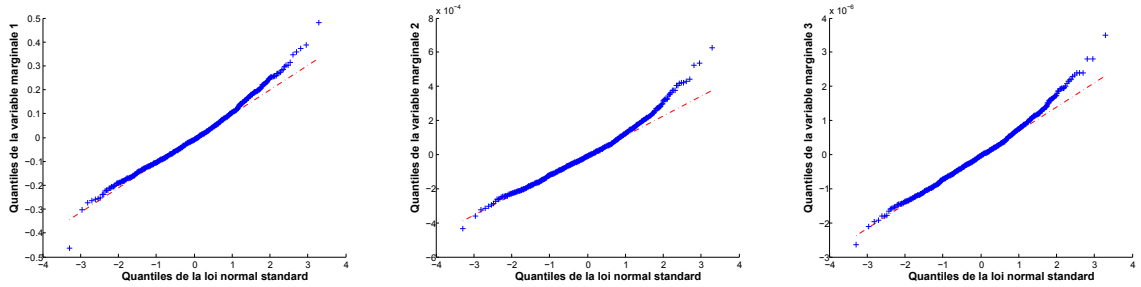


Figure 4.5 – 1000 instances générés par la simulation Monte Carlo et la loi multinormale.

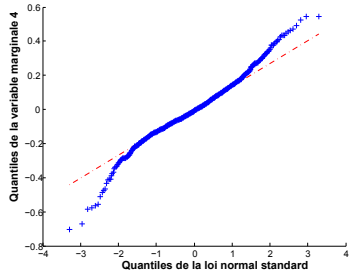
L'analyse de la Figure 4.5 montre que les instances, générés par la loi multinormale, suivent la même dispersion que celles des simulations de type Monte Carlo.



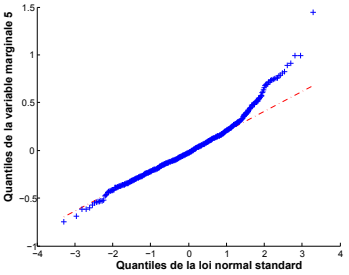
(a) Variable marginale 1

(b) Variable marginale 2

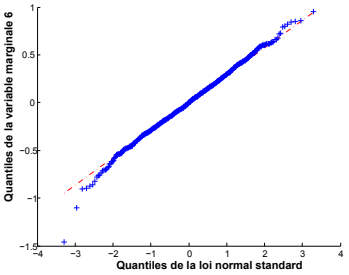
(c) Variable marginale 3



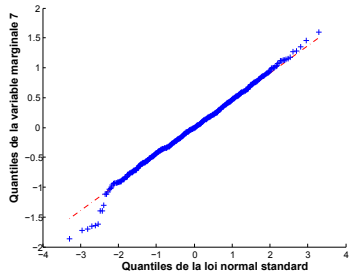
(d) Variable marginale 4



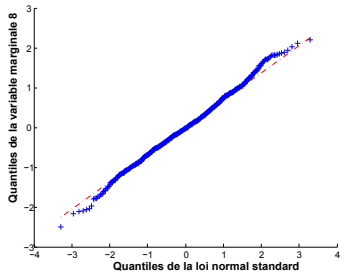
(e) Variable marginale 5



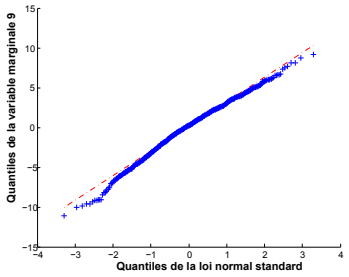
(f) Variable marginale 6



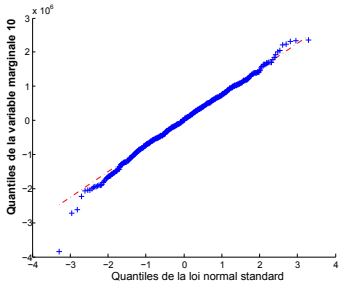
(g) Variable marginale 7



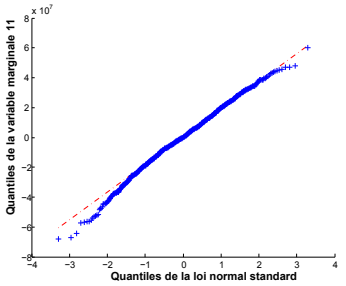
(h) Variable marginale 8



(i) Variable marginale 9



(j) Variable marginale 10



(k) Variable marginale 11

Figure 4.4 – Test de normalité de l'amplificateur opérationnel.

En outre, la Figure 4.6 montre la génération de 1000 et 1 million d'instances du circuit en utilisant la distribution multinormale. Il est clair qu'avec 1 million d'instances nous atteindrons une grande précision dans l'estimation des métriques de test.

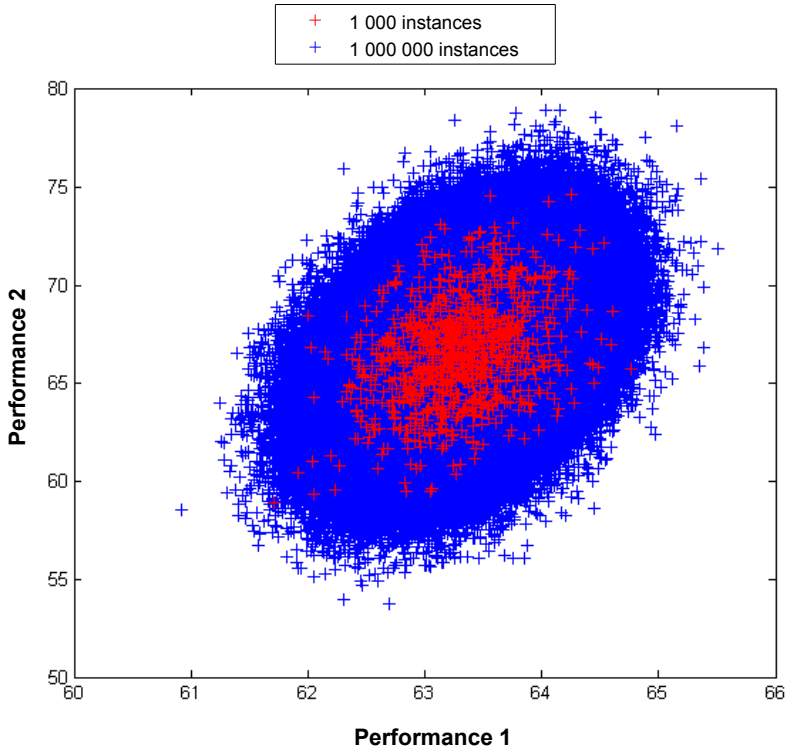


Figure 4.6 – Génération de 1000 et 1 million de circuits à partir de la distribution multinormale.

4.2.2.4 Estimation des paramètres du modèle par la méthode du bootstrap

Cette section a pour but de présenter le principe général de la méthode du bootstrap et quelques unes de ses applications dans l'estimation des paramètres de la loi multinormale. Nous présentons également l'application de la méthode dans la construction d'intervalles de confiance bootstrap de ces paramètres [52, 53].

Principe du bootstrap : les techniques de bootstrap interviennent lorsque le champ du problème considéré n'est pas couvert par des méthodes classiques d'estimation des paramètres ou lorsque les conditions d'application de ces dernières ne sont plus valables. L'idée est d'utiliser l'échantillon des observations pour permettre une inférence statistique plus fine. On réalise un certain nombre d'échantillons (qualifiés d'échantillons bootstrap) obtenus par tirage aléatoire non exhaustif d'observations de l'échantillon initial. Sur chacun des échantillons bootstrap, on estime un ou plusieurs paramètres du modèle. On obtient par conséquent une suite d'estimations pour chaque paramètre. Sous certaines conditions

de régularité, la théorie montre que la distribution de la suite des estimations obtenus converge vers la distribution réelle du paramètre recherché. Le principe de cette méthode est illustré sur la Figure 4.7, pour l'estimation d'un paramètre sur une population initiale de taille n en utilisant r échantillons bootstrap.

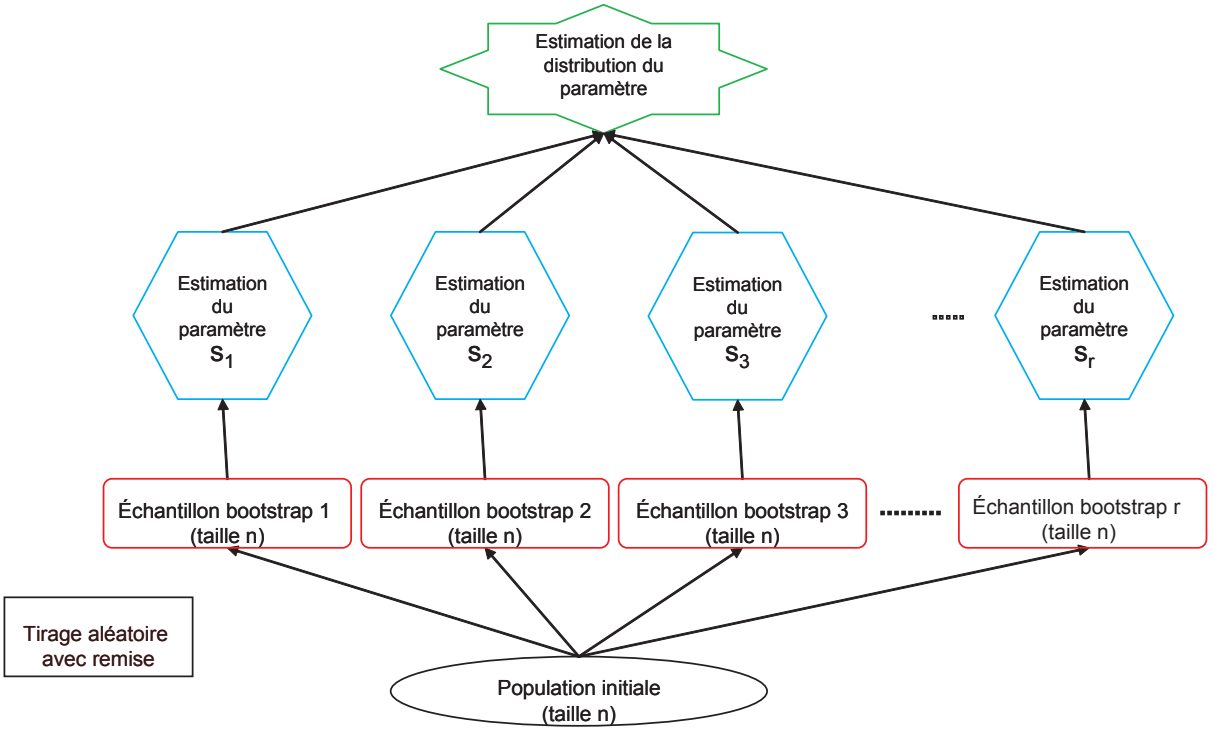


Figure 4.7 – Principe de la méthode du Bootstrap.

Cette dernière propriété théorique a donc favorisé l'utilisation des techniques bootstrap dans plusieurs problèmes classiques d'estimation :

1. L'estimation des erreurs des modèles.
2. L'estimation des intervalles de confiance.
3. L'estimation des biais des estimateurs.

Application de la méthode : l'application de cette méthode a été réalisée sur l'amplificateur différentiel de la Figure 4.3. Elle vise à réaliser deux objectifs :

1. Améliorer l'estimation du vecteur des moyennes et de la matrice de variance-covariance de la loi multinormale.
2. Construction des intervalles de confiance.

La moyenne : le Tableau 4.4 montre les résultats de l'application de la méthode du Bootstrap pour l'estimation des moyennes des différentes performances du circuit sous test et l'estimation des intervalles de confiance à 95%.

Performance	Moyenne sans Bootstrap	Moyenne avec Bootstrap	Intervalle de confiance à 95%
1. A_D	76.6132	76.6138	[76.5849 , 76.6444]
2. GBW_D	3.3006E+8	3.3008E+8	[3.2896E+8 , 3.3114E+8]
3. *Phase Margin* (PM)	63.3120	63.3117	[63.2815 , 63.3398]
4. $CMRR$	-42.7796	-42.7806	[-42.8443 , -42.7141]
5. $PSRR$ (G_{ND})	-30.0730	-30.0810	[-30.3 , -29.8438]
6. $PSRR$ (V_{DD})	-28.3270	-28.3328	[-28.5591 , -28.1066]
7. THD	66.5109	66.5097	[66.3484 , 66.6788]
8. *Current Consumption* (I_{DD})	0.0025	0.0025	[0.0025 , 0.0025]
9. *Intermodulation* (*Inter*)	67.6527	67.6503	[67.5763 , 67.7262]
10. $SR+$	7.2505E+7	7.2507E+7	[7.2156E+7 , 7.2873E+7]
12. *In Referred Noise* (*Noise*)	3.9288E-5	3.9288E-5	[0.3925E-4 , 0.3932E-4]

Tableau 4.4 – Tableau des moyennes estimées par le Bootstrap.

On remarque que les valeurs obtenues avec et sans le bootstrap sont les mêmes. Cette méthode n'apporte donc aucune précision sur l'estimation de la moyenne mais permet de construire des intervalles de confiance à 95%.

La variance : l'application de la méthode du Bootstrap pour l'estimation de la variance et la construction des intervalles de confiance est résumée dans le Tableau 4.5.

Performance	Variance sans Bootstrap	Variance avec Bootstrap	Intervalle de confiance à 95%
1. A_D	0.2342	0.2340	[0.2138 , 0.2557]
2. GBW_D	3.4253E+14	3.4157E+14	[3.1302E+14 , 3.7439E+14]
3. *Phase Margin* (PM)	0.2083	0.2073	[0.1896 , 0.2265]
4. $CMRR$	1.0152	1.0116	[0.9147 , 1.1136]
5. $PSRR$ (G_{ND})	12.6325	12.6207	[11.4986 , 13.8622]
6. $PSRR$ (V_{DD})	13.4323	13.4410	[12.2307 , 14.6996]
7. THD	7.0721	7.0629	[6.4418 , 7.7450]
8. *Current Consumption* (I_{DD})	0.5230E-7	0.5223E-7	[0.4702E-7 , 0.5758E-7]
9. *Intermodulation* (*Inter*)	1.5205	1.5201	[1.3722 , 1.6677]
10. $SR+$	3.9809E+13	3.9677E+13	[3.5877E+13 , 4.3821E+13]
12. *In Referred Noise* (*Noise*)	0.3099E-12	0.3094E-12	[0.2803E-12 , 0.3377E-12]

Tableau 4.5 – Tableau des variances estimées par le Bootstrap.

L'analyse des résultats du Tableau 4.5 montre que l'estimation de la variance est plus précise avec la méthode du bootstrap. En plus, elle permis la construction d'intervalles de confiance de la variance.

4.3 Modélisation non paramétrique

Le statisticien appliqué est souvent confronté au problème de l'estimation d'une distribution de probabilité à partir d'un échantillon $X_1, X_2, \ldots, X_n$. Le plus souvent, il est amené à faire des hypothèses compte tenu des informations extérieures sur la nature de cette distribution. Ces hypothèses simplificatrices permettent de limiter l'espace des

solutions à une estimation paramétrique. Une limitation importante de cette démarche classique est que le statisticien n'est que très rarement certain des hypothèses avancées dont la validité n'est vérifiée qu'à posteriori par des tests sur l'échantillon. Il arrive alors très fréquemment que ces tests donnent des résultats incertains, ne permettant pas une conclusion définitive. Ces problèmes peuvent être partiellement résolus par l'usage d'estimation non paramétrique de la densité de probabilité [54].

4.3.1 La méthode du noyau

La seule méthode simple et robuste permettant d'obtenir une estimation de la densité de probabilité d'une variable aléatoire réelle, ne nécessitant pas un choix multiple de paramètres, est la méthode du noyau de Parzen-Rosenblatt [46]. Si $X_1, X_2, \ldots, X_n$ est un échantillon de variables aléatoires indépendantes de même loi de densité f sur $\mathbb{R}$, on estime $f(x)$ par :

$$f_n(x) = \frac{1}{nh} \sum_{i=1}^{n} K(\frac{X_i - x}{h}) \tag{4.8}$$

où :

- $K(u)$ est appelé le noyau et vérifie les propriétés suivantes :
  1. $K(u) \geq 0$,
  2. $\int K(u)du = 1$,
  3. $|K(u)| \leq C$.
- h est un paramètre réel positif, appelé fenêtre ou largeur de la fenêtre.

Le choix de K ne présente pas, en général, une très grande importance, pourvu qu'il soit convenablement normalisé par h [55]. Les noyaux couramment utilisés sont résumés dans le Tableau 4.6, où $\mathbb{1}_{(p)}$ est la fonction indicatrice qui vaut 1 lorsque p est vrai, 0 sinon.

Noyau	$K(u)$
Uniforme	$\frac{1}{2}\mathbb{1}_{\{\lvert u\rvert \leq 1\}}$
Triangle	$(1 - \lvert u\rvert)\mathbb{1}_{\{\lvert u\rvert \leq 1\}}$
Epanechnikov	$\frac{3}{4}(1 - u^2)\mathbb{1}_{\{\lvert u\rvert \leq 1\}}$
Quadratique	$\frac{15}{16}(1 - u^2)^2\mathbb{1}_{\{\lvert u\rvert \leq 1\}}$
Triweight	$\frac{35}{32}(1 - u^2)^3\mathbb{1}_{\{\lvert u\rvert \leq 1\}}$
Gaussien	$\frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}u^2)$
Cosine	$\frac{\pi}{4} \cos(\frac{\pi}{2}u)\mathbb{1}_{\{\lvert u\rvert \leq 1\}}$

Tableau 4.6 – Tableau des principaux noyaux.

Le choix de h présente une importance considérable pour l'efficacité de l'estimation. Un choix de h trop petit implique la présence de fluctuations aléatoires importantes, un choix de h trop grand élimine les aléas, mais introduit des biais de lissage importants comme

illustré sur la Figure 4.8. Plusieurs méthodes ont été proposées pour le choix optimal de la valeur de la fenêtre h [55, 46].

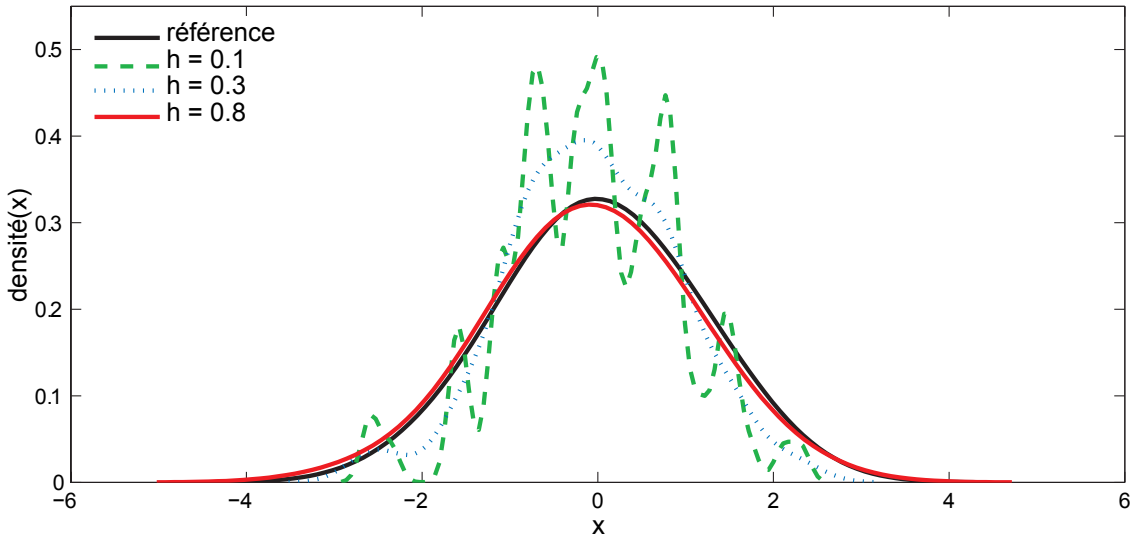


Figure 4.8 – Estimation par la méthode du noyau pour différentes valeurs de la largeur de fenêtre h.

Pour assurer un choix optimal des paramètres de la fonction de densité de probabilité par la méthode du noyau, on va s'imposer un test statistique en plus des méthodes optimales de choix de la largeur de la fenêtre afin de vérifier que l'échantillon issu de l'estimation par la méthode du noyau correspond bien à l'échantillon de départ. On utilise le test de Kolmogorv-Smirnov.

4.3.2 Le test de Kolmogorov-Smirnov

En statistique, le test de Kolmogorov-Smirnov [56] est un test d'hypothèse utilisé pour déterminer si deux lois continues F et G sont égales. Pour cela, on prend un m-échantillon $X_1, X_2, \ldots, X_m$ de loi F et un n-échantillon $X_1, X_2, \ldots, X_n$ de loi G.

La fonction de répartition empirique F_m pour m observations x_i est définie ainsi :

$$F_m(x) = \frac{1}{m} \sum_{i=1}^{m} (\delta_{x_i} \leq x)$$

avec

$$\delta_{x_i} \leq x = \begin{cases} 1, & \text{si } x_i \leq x\,; \\ 0, & \text{sinon.} \end{cases}$$

Le test de $H_0 : F = G$ (hypothèse nulle) contre $H_1 : \{F$ différent de $G\}$ a pour région de rejet au niveau α, $\{D_{m,n} \geq C_\alpha\}$ où

$$D_{m,n} = \sup_{x \in \mathbb{R}} |F_m(x) - G_n(x)|$$

sous l'hypothèse nulle. La constante C_α est indépendante et est donnée par la table de Kolmogorov-Smirnov.

4.3.3 La méthode du noyau multidimensionnel

La méthode du noyau peut facilement être généralisée au cas multidimensionnel (d dimensions). La densité de probabilité pour un noyau K et une largeur de fenêtre h est définie par [46]

$$f_n(x) = \frac{1}{nh^d} \sum_{i=1}^{n} K(\frac{x - X_i}{h}) \tag{4.9}$$

Le noyau $K(x)$ est défini pour un x à d dimensions et doit satisfaire

$$\int_{\mathbb{R}^d} K(x)dx = 1. \tag{4.10}$$

Par exemple, le noyau multinormal standard s'écrit sous la forme

$$K(x) = (2\pi)^{\frac{-d}{2}} \exp(-\frac{1}{2}x^T x). \tag{4.11}$$

4.3.4 Exemple d'application

La modélisation statistique par la méthode du noyau a été réalisée sur un amplificateur faible bruit LNA [5]. Ce dernier, conçu sous la technologie BiCMOS [6] 0.25μm de STMicroelectronics, est illustré sur la Figure 4.9.

Le circuit doit satisfaire 5 performances dont les spécifications sont données dans le Tableau 4.7.

	Spécification	
Performance	a_1	a_2
1. NF	$-\infty$	1.3 dB
2. S_{11}	$-\infty$	-9 dB
3. Gain	17 dB	$+\infty$
4. 1-dB CP	-11.3 dBm	$+\infty$
5. IIP_3	-5.1 dBm	$+\infty$

Tableau 4.7 – Les spécifications du LNA.

La simulation Monte Carlo est réalisée sur le circuit sous test pour générer 1000 instances du LNA. Le test de Kolmogorov-Smirnov est utilisé pour un meilleur choix des paramètres de la méthode du noyau. Il s'agit du noyau multinormal (4.11) et de la fenêtre calculée par la méthode empirique [7] [46], implémenté dans Matlab [57].

5. Low Noise Amplifier
6. Bipolar Complementary Metal Oxide Semi-conductor
7. Rule of Thumb Method

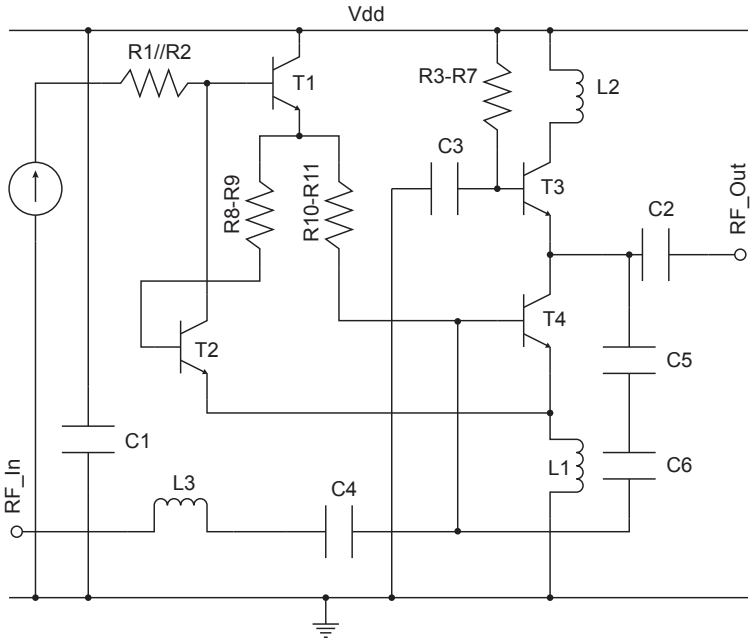


Figure 4.9 – Le circuit sous test (LNA).

La Figure 4.10 montre la distribution de 3 performances dans le cas de 1000 instances générées par une simulation Monte Carlo et 1000 instances issues de la densité de probabilité construite avec la méthode du noyau. Les deux distributions sont approximativement confondues et des résultats similaires sont obtenus pour les autres performances.

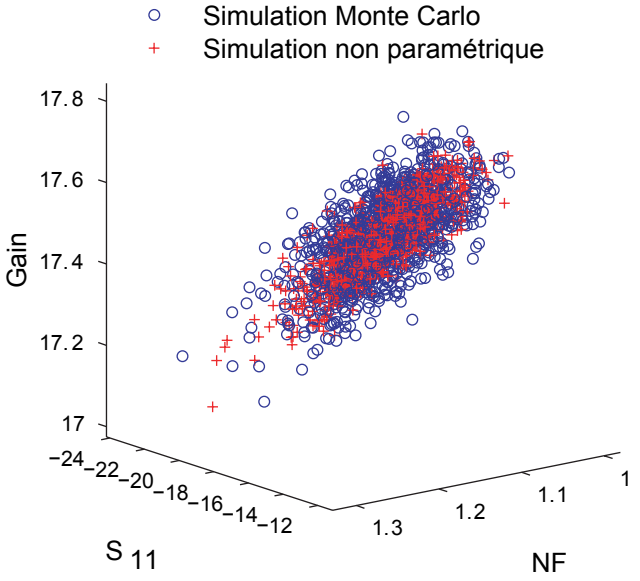


Figure 4.10 – Distribution de 1000 circuits générés par la simulation Monte Carlo et la méthode du noyau.

4.4 Modélisation basée sur les copules

L'estimation de distribution multivariée n'est pas facile, alors que l'estimation univariée pose moins de problème. Ainsi dans notre cas, les distributions des marginales peuvent être facilement estimées par la méthode du noyau [46]. La notion de copule [45, 58, 59] utilise les estimations des marginales pour reconstruire la distribution multivariée. Les copules capturent la structure de dépendance et permettent de séparer la modélisation de celle-ci de la modélisation des distributions marginales. En isolant cette structure de dépendance, il est alors possible de déterminer la loi multivariée originelle, car composée de différentes lois marginales.

La mesure de dépendance la plus utilisée est la corrélation linéaire. Cet indicateur est performant quand la relation de dépendance est linéaire et évolue dans un univers gaussien. Ce cas de figure n'est pas toujours vérifié. Pour remédier à cela, on fait appel à d'autres indicateurs de dépendance basés sur les discordances et concordances observés sur un échantillon. Nous utilisons alors des coefficients de corrélation non linéaires et non paramétriques, comme :

- τ de Kendall ;
- ρ_s de Spearman.

Ce sont de bons indicateurs globaux de la dépendance entre variables aléatoires. En outre, ils sont compris entre -1 et 1 comme le coefficient de corrélation linéaire : une valeur de 1 par exemple signifie une concordance parfaite. Les indicateurs de dépendance (corrélation linéaire, τ de Kendall et ρ_s de Spearman) seront définis comme paramètres de la copule dans le cas paramétrique. Dans le cas normal bivarié, le τ de Kendall ou le ρ_s de Spearman sont liés à la corrélation linéaire ρ par :

$$\tau = \frac{2}{\pi}\arcsin(\rho) \quad \text{ou} \quad \rho = \sin(\tau\frac{\pi}{2}) \tag{4.12}$$

$$\rho_s = \frac{6}{\pi}\arcsin(\frac{\rho}{2}) \quad \text{ou} \quad \rho = 2\sin(\rho_s\frac{\pi}{6}) \tag{4.13}$$

4.4.1 Définition

Nous nous limitons dans cette section à l'étude des copules bivariées, dont les résultats sont facilement généralisables aux copules multivariées.

Définition 4.4.1. La copule bivariée C fonction de $[0,1]^2 \to [0,1]$ est définie par les caractéristiques suivantes :

1. $C(u,0) = C(0,u) = 0\ \forall u \in [0,1]$.
2. $C(u,1) = C(1,u) = u\ \forall u \in [0,1]$: les distributions marginales sont des lois uniformes.

3. C est 2-croissante[8] : $C(v_1, v_2) - C(v_1, u_2) - C(u_1, v_2) + C(u_1, u2) \geq 0 \, \forall (u_1, u_2) \in [0,1]^2, (v_1, v_2) \in [0,1]^2$ tel que $0 \leq u1 \leq v_1 \leq 1$ et $0 \leq u_2 \leq v_2 \leq 1$.

Soient U_1 et U_2 deux variables aléatoires uniformes sur $[0,1]$[9], alors on a $C(u_1, u_2) = P(U_1 \leq u_1, U_2 \leq u_2) \, \forall (u_1, u_2) \in [0,1]^2$. Cette définition assure donc que la copule est une distribution de probabilité avec des distributions marginales uniformes.

4.4.2 Théorème de Sklar

Une copule est déterminée soit à partir de la définition 4.4.1, soit à l'aide d'une loi bivariée existante. Dans ce cas, on fait appel au théorème de Sklar [60]. Ce théorème précise le lien défini par la copule C, déterminé à partir de la distribution conjointe F, entre les fonctions de répartition marginales univariées F_1 et F_2 et la distribution complète bivariée F.

Théorème 4.4.1. Soit F une distribution bivariée de marginales F_1 et F_2. La copule C associée à F s'écrit :

$$\begin{aligned} C(u_1, u_2) &= C(F_1(x_1), F_2(x_2)) \\ &= F(F_1^{-1}(u_1), F_2^{-1}(u_2)) \\ &= F(x_1, x_2) \end{aligned}$$

C est unique lorsque les distributions marginales F_1 et F_2 sont continues.

La densité f d'une loi bivariée peut s'écrire aussi en fonction de la densité c de la copule associée et des densités des marginales f_1 et f_2 :

$$f(x_1, x_2) = c(F_1(x_1), F_2(x_2)) \times f_1(x_1) \times f_2(x_2) \tag{4.14}$$

La modélisation d'une copule signifie, en même temps, la modélisation des distributions marginales et de la loi conjointe [58]. Alors, la validation d'une modélisation par une copule demande la validation de l'ajustement des lois marginales ainsi que de la copule elle-même. Plusieurs méthodes sont proposées dans la littérature mais aucune d'entre elles n'est considéré comme étant la meilleure [58]. En général, toutes ces méthodes ramènent le problème à un test univarié par une transformation de la distribution multivariée. Ainsi, Malvergne et all [61] utilise le test de Kolmogorov-Smirnov [56] pour tester le résultat de la transformation avec une distribution χ^2 (khi-deux). Dans notre travail, nous avons adopté le même principe que dans le cas de la modélisation non paramétrique (voir le section 4.3.2). Il s'agit de la validation par le test de Kolmogorov-Smirnov effectué entre

8. 2-increasing

9. Une loi uniforme U sur $[0,1]$ a pour fonction de répartition : $P(U \leq u) = \begin{cases} 0 & \text{si } u \leq 0 \\ u & \text{si } 0 \leq u \leq 1 \\ 1 & \text{si } u \geq 1 \end{cases}$

les données originelles et les nouvelles simulations obtenues après rééchantillonnage de la copule.

4.4.3 Exemples de copules

Il existe de nombreuses familles de copules suivant la dépendance qu'elles représentent. Une famille a plusieurs paramètres se rapportant à la forme et l'intensité de la dépendance. Dans cette section, on va détailler 3 grandes familles de copules : gaussienne, Student et archimédiennes.

4.4.3.1 Copule gaussienne

Définition 4.4.2. La copule gaussienne bivariée est définie de la façon suivante : $C(u_1, u_2, \rho) = \Phi_\rho(\phi^{-1}(u_1), \phi^{-1}(u_2))$ où ρ est le coefficient de corrélation, ϕ la distribution normale standard et Φ_ρ la distribution normale bivariée standard, fonction de ρ.

La Figure 4.11 permet de visualiser la densité de la copule gaussienne pour différentes valeurs du coefficient de corrélation ρ.

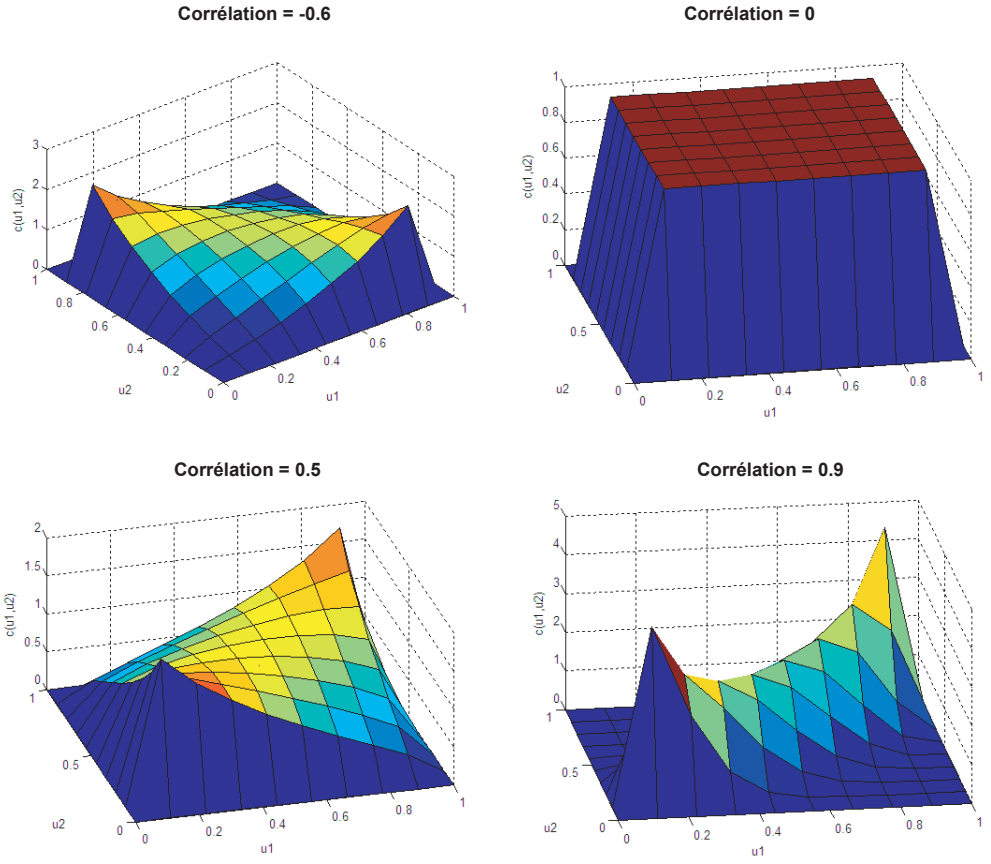


Figure 4.11 – Densité de quatre copules gaussiennes bivariées pour différentes valeurs de ρ.

4.4.3.2 Copule de Student

La copule de Student est extraite de la même manière que la copule gaussienne mais cette fois-ci à partir de la distribution de Student bivariée.

Définition 4.4.3. La copule de Student bivariée est définie de la façon suivante : $C(u_1, u_2, \rho, k) = T_{\rho,k}(T_k^{-1}(u_1), T_k^{-1}(u_2))$ avec ρ le coefficient de corrélation, T_k la distribution de Student standard et $T_{\rho,k}$ la distribution de Student bivariée standard, fonction de ρ et de degré de liberté k.

La Figure 4.12 montre la densité de la copule de Student pour différentes valeurs du coefficient de corrélation ρ et un degré de liberté $k = 1$.

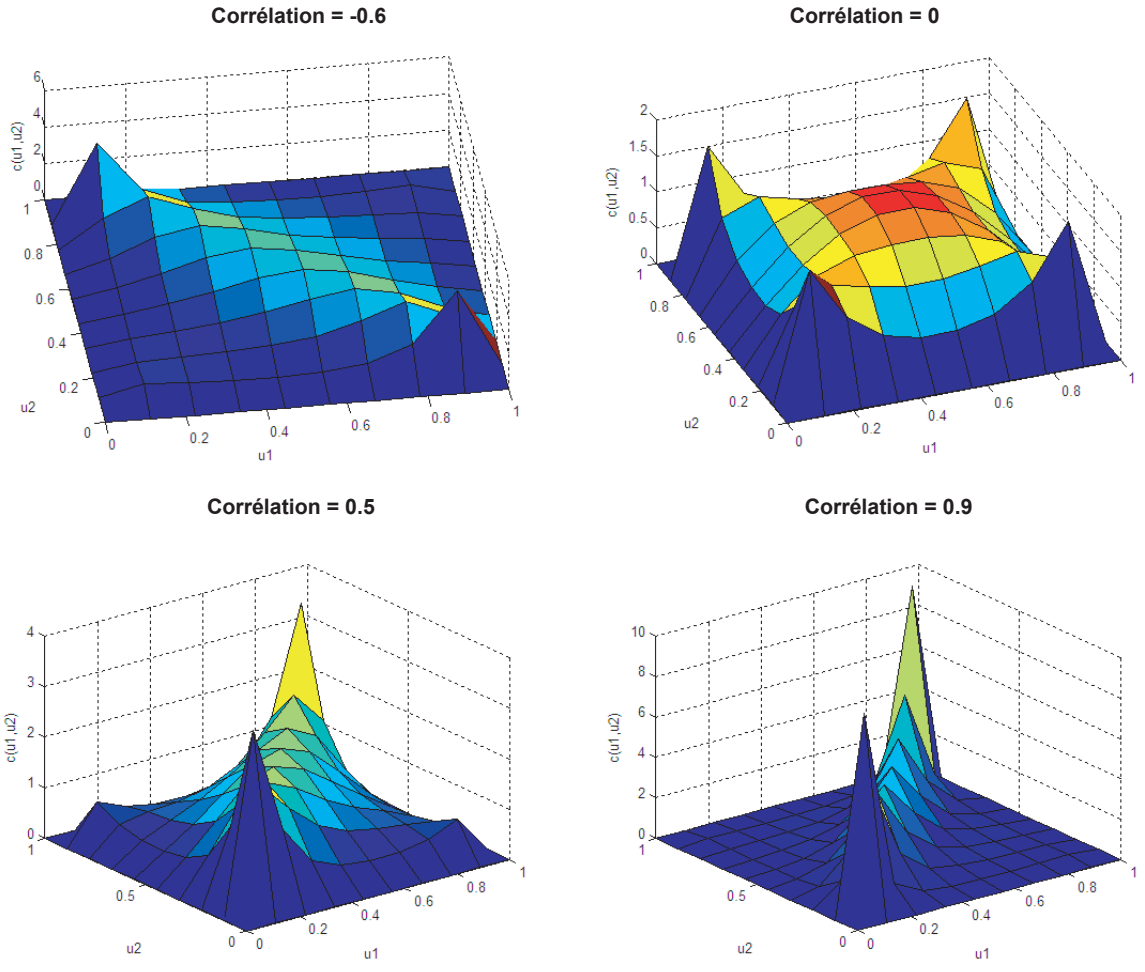


Figure 4.12 – Densité de quatre copules de Student bivariées pour différentes valeurs de ρ et un degré de liberté $k = 1$.

4.4.3.3 Copule archimédienne

Définition 4.4.4. Les copules archimédiennes sont définies de la manière suivante : $C(u_1, u_2) = \begin{cases} \varphi^{-1}(\varphi(u_1) + \varphi(u_2))) & \text{si } \varphi(u_1) + \varphi(u_2) \leq \varphi(0) \\ 0 & \text{sinon} \end{cases}$
avec φ vérifie $\varphi(1) = 0$, $\varphi'(u) < 0$ et $\varphi''(u) > 0$ pour tout $0 \leq u \leq 1$. φ est appelée la fonction génératrice de la copule.

Le tau de Kendall τ est égal pour les copules archimédiennes à :

$$\tau = 1 + 4 \int_0^1 \frac{\varphi(u)}{\varphi'(u)} du \tag{4.15}$$

En notant $\tilde{u} = -\ln u$, le Tableau 4.8 montre quelques exemples de copules archimédiennes bivariées :

Nom	Générateur	Copule bivariée
Clayton ($\theta > 0$)	$u^{-\theta} - 1$	$(u_1^{-\theta} + u_2^{-\theta} - 1)^{-1/\theta}$
Gumbel ($\theta \geq 1$)	$(\ln u)^\theta$	$\exp(-(\tilde{u}_1^\theta + \tilde{u}_2^\theta)^{1/\theta})$
Frank ($\theta \neq 0$)	$-\ln \frac{e^{-\theta u} - 1}{e^{-\theta} - 1}$	$-\frac{1}{\theta} \ln \left(1 + \frac{(e^{-\theta u_1} - 1)(e^{-\theta u_2} - 1)}{e^{-\theta} - 1}\right)$

Tableau 4.8 – Exemple de copules archimédiennes bivariées.

La Figure 4.13 illustre les densités de trois copules archimédiennes bivariées de paramètre $\theta = 4$.

Pour la copule de Gumbel, le τ de Kendall en fonction de θ est donné par :

$$\tau = 1 - \frac{1}{\theta} \tag{4.16}$$

4.4.4 Méthode de simulation

Simuler une copule bivariée C signifie simuler les arguments u_1 et u_2 de cette fonction tirés d'un vecteur aléatoire de loi uniforme $U = (U_1, U_2)$ de distribution C. Ceci permet alors de déterminer la simulation d'un vecteur aléatoire $X = (X_1, X_2)$ dont la structure de dépendance est définie par C et dont nous définissons des distributions marginales particulières F_1 et F_2. Nous utiliserons la transformation $X = (F_1^{-1}(U_1), F_2^{-1}(U_2))$ donnée par le théorème 4.4.1. La plupart des logiciels de statistique proposent des générateurs de nombres aléatoires suivant les copules les plus utilisées (gaussienne, Student et archimédienne).

Les principales étapes de la simulation d'un échantillon par les copules sont :

1. Ajustement de chaque variable par une loi paramétrique usuelle.
2. Utiliser les fonctions de répartition des lois marginales pour construire les lois uniformes U sur $[0, 1]$.
3. Construire la copule correspondante à U.

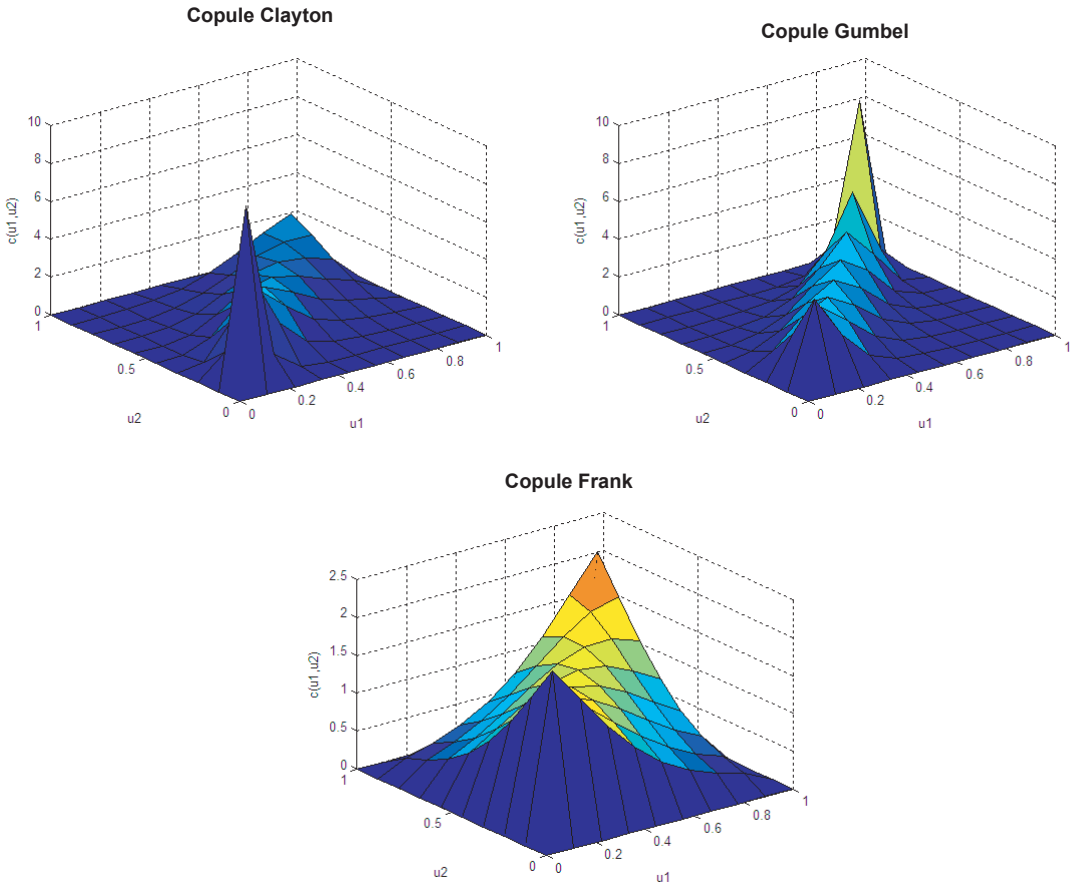


Figure 4.13 – Densité de trois copules archimédiennes bivariées de paramètre $\theta = 4$.

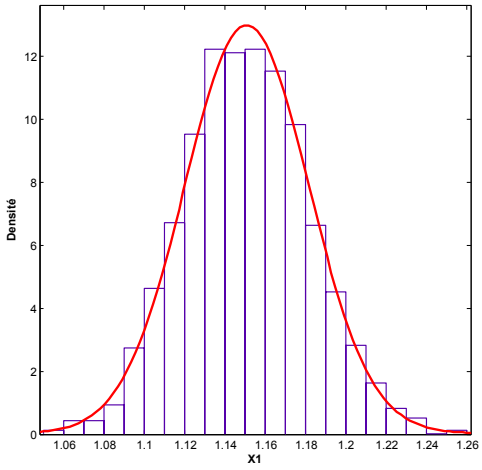
4. Générer de nouvelles simulations de la copule.
5. utiliser les fonctions inverses des fonctions de répartitions F^{-1} pour calculer les simulations X.

Ces étapes permettent à partir d'un petit échantillon, issu à l'étape 1 de la production ou bien de la simulation Monte Carlo, de rééchantillonner plusieurs millions d'instances de la copule à l'étape 4, puis d'utiliser la méthode de l'inverse pour obtenir un grand échantillon du circuit sous test à la dernière étape.

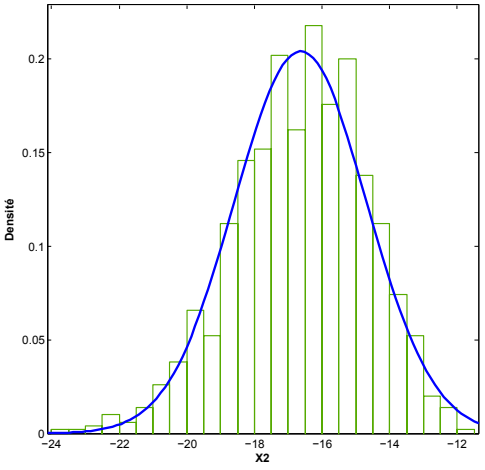
4.4.5 Exemple d'application

L'application a été réalisée sur l'amplificateur faible bruit LNA illustré sur la Figure 4.9 de la section précédente et en utilisant les même spécifications du Tableau 4.7. La première étape de l'application des copules consiste à ajuster les lois marginales des 5 performances. Les résultats de cette ajustement sont illustrés sur la Figure 4.14 avec les paramètres d'ajustement résumés dans le Tableau 4.9.

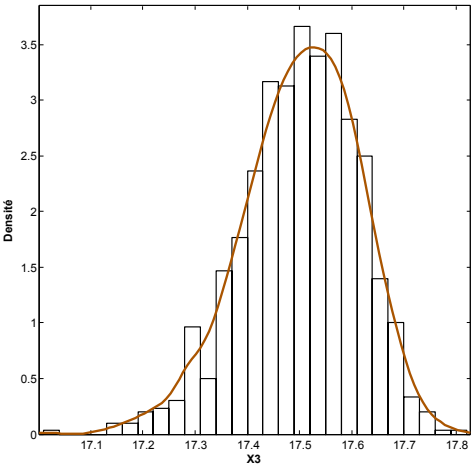
La copule gaussienne a permis de générer un échantillon de 1 million d'instances du LNA illustré sur la Figure 4.15.



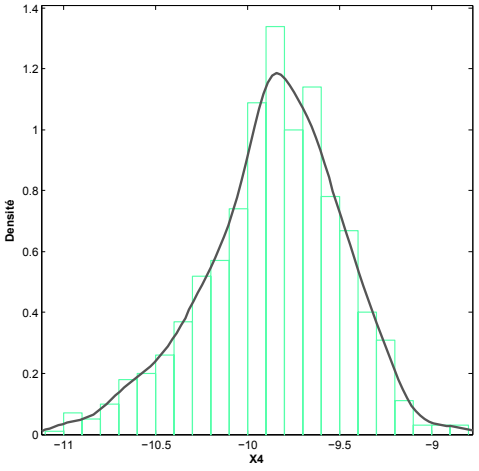
(a) NF



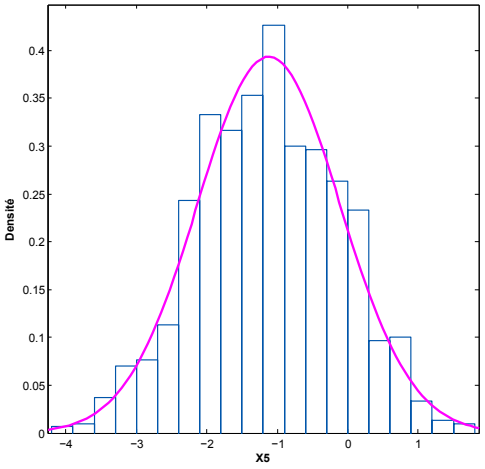
(b) S_{11}



(c) Gain

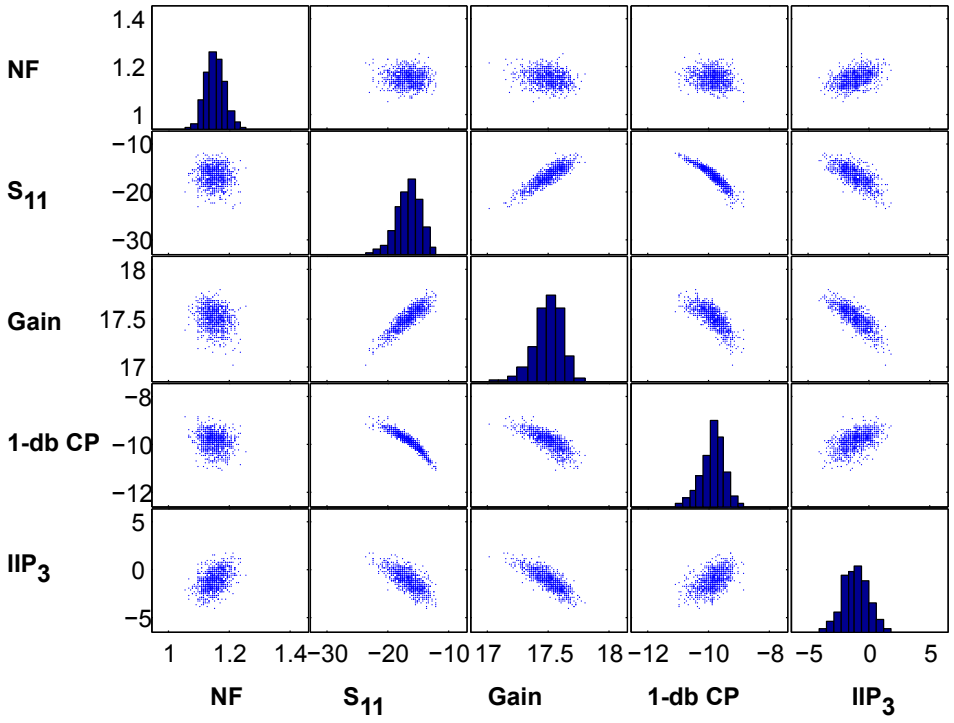


(d) 1-dB CP

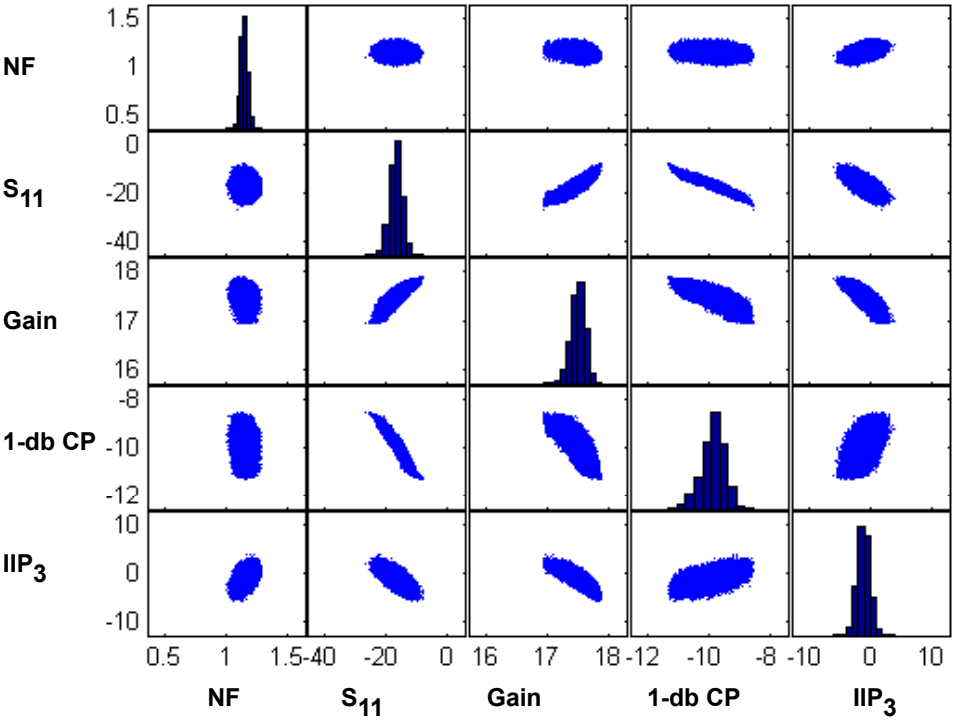


(e) IIP_3

Figure 4.14 – Ajustement des lois marginales du LNA.



(a) Données (1000 instances)



(b) Simulation (1 million d'instances)

Figure 4.15 – Génération de 1 million d'instance du LNA.

Performance	Variable	Loi d'justement	Paramètres
NF	$X1$	Normale	$(\mu = 1.15, \sigma = 0.03)$
S_{11}	$X2$	Normale	$(\mu = -16.633, \sigma = 1.953)$
Gain	$X3$	KDE	Noyau=Gaussien, $h = 0.029$
1-dB CP	$X4$	KDE	Noyau=Gaussien, $h = 0.089$
IIP_3	$X5$	Normale	$(\mu = -1.121, \sigma = 1.013)$

Tableau 4.9 – Paramètres d'ajustement des densités marginales du LNA.

4.5 Conclusion

Nous avons défini dans ce chapitre trois différentes manières de construire un modèle statistique du circuit sous test. Si chaque performance du circuit suit une loi normale et que les corrélations entre les performances sont linéaires, alors le modèle multinormal peut être plus adéquat. Toutefois, si l'une de ces deux conditions n'est pas remplie, on pourra utiliser un modèle non paramétrique, plus générique et qui s'adapte à tous les cas de figure. Cependant, ce dernier modèle présente l'inconvénient de la dimension. En effet, dès qu'on travaille avec un nombre important de performances, le modèle perd en précision et plus de données de départ sont nécessaires pour construire un bon modèle. Une solution robuste de modélisation est l'utilisation des copules pour extraire la structure de dépendance d'une distribution conjointe et ainsi de séparer la structure de dépendance du comportement marginal. Une fois le modèle validé par un test statistique qui assure une quantification du risque, on passe à l'étape de simulation du modèle qui permet de générer un échantillon de grande taille en peu de temps. Ceci permet de pallier les inconvénients de la simulation Monte Carlo en termes de taille des échantillons, temps de simulation et précision dans l'estimation des métriques de test.

Chapitre 5

Méthode d'ordonnancement et de réduction de tests

5.1 Introduction

Dans ce chapitre, on détaillera les différentes méthodes utilisées dans l'ordonnancement des tests. Une fois le modèle statistique du circuit sous test construit suivant les différentes méthodes de modélisation présentées dans le chapitre précèdent, le modèle statistique est rééchantillonné pour générer une grande population de circuits (plusieurs millions) en peu de temps. La phase suivante de notre approche consiste à utiliser cette population de circuits pour l'ordonnancement des tests. Le but est d'ordonner les tests suivant l'estimation du taux de défauts paramétriques en utilisant la modélisation statistique. Cet ordre est obtenu en inversant l'ordre d'élimination des tests obtenu par minimisation du taux de défauts. Au final, nous obtiendrons un ordonnancement des tests capable de détecter les circuits défectueux au plus tôt, ainsi qu'un ordre d'élimination des tests avec une estimation de l'erreur commise sous forme d'intervalles de confiance du taux de défauts. Ce dernier résultat proposera l'ensemble des tests à éliminer en fonction de l'erreur tolérée sur le taux de défauts.

Pour commencer, nous présentons quelques heuristiques simples susceptibles d'être utilisées par l'ingénieur de test et employant peu de données mesurées sur les circuits fonctionnels et ne consommant pas trop de ressources de calcul. Le résultat de ces heuristiques nous servira par la suite de référence par rapport à la méthode d'ordonnancement présentée dans ce travail. Ensuite, nous détaillerons le principe de la méthode d'ordonnancement à travers son évolution d'une simple heuristique d'élimination des tests vers une méthode d'ordonnancement plus générale, applicable sur tous les types de circuits et adaptative vis-à-vis des données disponibles. La dernière section proposera une méthode de décomposition des tests d'un circuit complexe en sous-ensembles de test plus faciles à modéliser et à ordonnancer.

5.2 Les heuristiques simples

Il est facile de proposer un ordonnancement des tests d'un circuit en se basant sur un principe évident comme la réduction du temps de test, du coût de test ou des erreurs de test. Mais comment être sûr que cet ordre est optimal ou approche l'ordre optimal. Et quel est le coût de la mise en ouvre de cet ordonnancement en termes de temps et moyens de calcul ? En effet, comme on ne connait pas l'ordre réel et optimal des tests, il est difficile de juger de la qualité d'un ordonnancement obtenu par une méthode donnée.

Une première approche de validation d'un ordonnancement est d'utiliser un circuit avec un ordonnancement des tests connus et optimal. Une méthode de construire un tel circuit (artificiel) est d'utiliser un modèle statistique connu par exemple le modèle multi-normal. Puis, en fixant les limites des spécifications on pourra définir un ordre optimal des tests qui servira d'ordre de référence. Cette approche sera proposée comme un exemple d'application de la méthode de décomposition (voir la section 5.5).

Dans cette section, on propose deux approches ou heuristiques simples et évidentes pour ordonner les tests d'un circuit. Ces heuristiques ne s'appuient que sur peu de données issues des circuits fonctionnels. Ces circuits sont faciles à obtenir par la simulation Monte Carlo ou pendant la production des premiers lots du circuit. La première heuristique est basée sur les corrélations entre les tests et la seconde heuristique est basée sur la capabilité[1].

5.2.1 Heuristique simple basée sur la corrélation

Cette heuristique simple n'utilise pas la modélisation statistique mais se base sur les corrélations entre les tests pour les ordonner. Un test est ordonné suivant la somme de ses corrélations avec les autres tests en valeur absolue. Le test ayant la plus grande somme des corrélations sera éliminé en premier et ainsi de suite pour les autres tests restants.

5.2.1.1 Exemple

Prenons l'exemple du LNA de la Figure 4.9 avec 5 tests, la matrice des corrélations C en valeur absolue est :

$$|C| = \begin{bmatrix} 1 & 0,051768765 & 0,214633679 & 0,157136337 & 0,431353649 \\ 0,051768765 & 1 & 0,875511768 & 0,936102654 & 0,707008374 \\ 0,214633679 & 0,875511768 & 1 & 0,762515344 & 0,80881373 \\ 0,157136337 & 0,936102654 & 0,762515344 & 1 & 0,518533587 \\ 0,431353649 & 0,707008374 & 0,80881373 & 0,518533587 & 1 \end{bmatrix} \quad (5.1)$$

1. Process capability

Alors, la somme des corrélations s par ligne est :

$$s_0 = \begin{bmatrix} 1,85489243 \\ 3,570391561 \\ 3,661474521 \\ 3,374287923 \\ 3,465709341 \end{bmatrix} \tag{5.2}$$

Le test numéro 3 a la plus grande somme des corrélations donc ce test doit être éliminé en premier. Ce choix est motivé par le fait que l'information disponible sur le test 3 éliminé est contenue dans les tests restants $\{1, 2, 4, 5\}$ car fortement corrélée avec eux. Puis, on élimine le test numéro 2 et ainsi de suite suivant un ordre décroissant de la somme des corrélations. Le résultat final est résumé dans le Tableau 5.1.

Ordre d'élimination	**Ordonnancement**
3	4
2	1
5	5
1	2
4	3

Tableau 5.1 – Ordonnancement des tests du LNA suivant l'heuristique basée sur la corrélation.

Ce résultat montre que les tests doivent être appliqués dans l'ordre : 4, 1, 5, 2 et 3 pour détecter au plus tôt les circuits défectueux. En plus, si on devait éliminer un seul test ça serait le numéro 3, puis 2, 5 et 1.

5.2.2 Heuristique de la capabilité

L'heuristique simple basée sur les corrélations observe bien les relations entre les tests deux à deux mais ne tient pas compte des spécifications. Or les limites de test peuvent influencer considérablement la vérification ou non d'un test. Avec des limites serrées le test a plus de chance d'être violé qu'avec des limites éloignées. Partant de ce constat, nous avons proposé une heuristique simple basée sur la capabilité qui tient compte des limites des spécifications.

5.2.2.1 Définition de la capabilité

La capabilité d'un procédé de production est l'adéquation d'une machine ou d'un procédé à réaliser une performance demandée. Elle permet de mesurer la capacité d'une machine ou d'un procédé à réaliser des pièces dans l'intervalle des spécifications ou de tolérance ([LI [2] ; LS [3]]) défini dans le cahier des charges [62]. Un procédé est dit capable si

2. Limite Inférieure
3. Limite Supérieure

toutes ses mesures sont contenues dans leurs intervalles de spécification. Ceci est représenté sur la Figure 5.1.

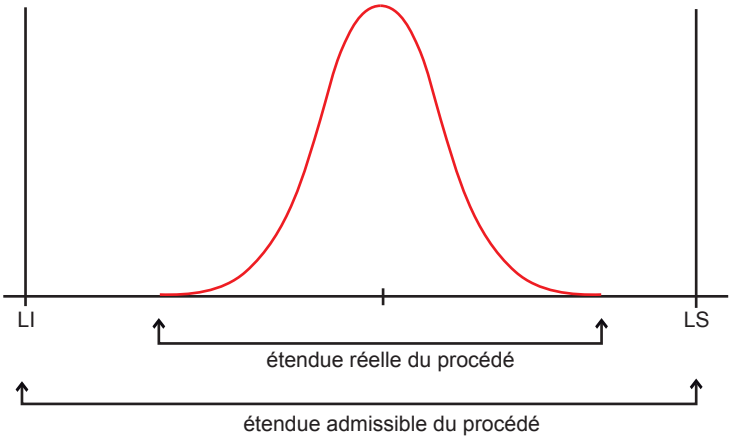


Figure 5.1 – Capabilité d'un procédé.

Il y a plusieurs statistiques qui peuvent être employées pour mesurer la capabilité d'un procédé : C_p, C_{pk}. Les statistiques C_p et C_{pk}, appelés aussi indices de capabilité, supposent que la population des données est normalement distribuée. Si μ et σ sont, respectivement, la moyenne et l'écart-type des données, LS la limite supérieure des spécifications et LI sa limite inférieure, alors les indices de capabilité de la population sont définis comme suit :

$$C_p = \frac{LS - LI}{6\sigma} \tag{5.3}$$

$$C_{pk} = \min\left[\frac{LS - \mu}{3\sigma}, \frac{\mu - LI}{3\sigma}\right] \tag{5.4}$$

N'ayant pas les valeurs exactes de μ et σ, on peut utiliser leurs estimateurs respectifs $\bar{x}$ et s pour obtenir les estimateurs des indices de capabilité :

$$\hat{C_p} = \frac{LS - LI}{6s} \tag{5.5}$$

$$\hat{C_{pk}} = \min\left[\frac{LS - \bar{x}}{3s}, \frac{\bar{x} - LI}{3s}\right] \tag{5.6}$$

L'estimateur de C_{pk} peut être écrit en fonction de C_p comme $C_{pk} = C_p(1 - k)$, où k est la distance entre la médiane m des spécifications et la moyenne μ.

$$k = \frac{|m - \mu|}{(LS - LI)/2} \; 0 \leq k \leq 1 \tag{5.7}$$

La qualité 6σ signifie que si on a un écart-type de 6σ entre la moyenne du procédé et la limite de spécification la plus proche, pratiquement aucun articles ne parviendra à

violer les spécifications. Pour atteindre cette qualité (6σ), C_p doit être supérieur à 2 ou bien C_{pk} supérieur à 1,5 [62] sinon il y a une instabilité du procédé.

5.2.2.2 Principe de l'heuristique

Comme la valeur de C_p n'est pas définie quand une des limites est à l'infini (ce qui est souvent le cas pour les circuits analogiques et RF), on va utiliser le C_{pk} pour l'ordonnancement des tests. Le principe de l'heuristique de la capabilité est que plus la valeur du C_{pk} est grande moins le test à de chances d'être violé. L'ordonnancement des tests correspond donc à l'ordre croissant des valeurs de C_{pk}.

5.2.2.3 Exemple

Prenons le même cas d'étude de la section précédente. Nous allons appliquer l'heuristique de la capabilité sur le LNA, comportant 5 tests. Les résultats d'application de l'heuristique sont résumés dans le Tableau 5.2.

N° du test	*LI*	*LS*	C_{pk}	**Ordre d'élimination**	**Ordonnancement**
1	$-\infty$	1,2734	1,327212023	1	4
2	$-\infty$	-8,9318	1,31389849	5	2
3	17,0663	$+\infty$	1,317263092	3	3
4	-11,3132	$+\infty$	1,311767828	2	5
5	-5,1391	$+\infty$	1,320751636	4	1

Tableau 5.2 – Ordonnancement des tests du LNA suivant l'heuristique simple du testeur.

Une analyse des résultats du Tableau 5.2, montre que le premier test à mesurer est le test numéro 4 car il possède la valeur minimale de $C_{pk} = 1,311767828$. Il est ensuite suivie du test numéro 2 et ainsi de suite jusqu'au dernier test qui est le numéro 1 car possédant la plus grande valeur de $C_{pk} = 1,320751636$. Une comparaison de cet ordonnancement des tests avec celui obtenu avec l'heuristique basée sur la corrélation montre que les deux heuristiques s'accordent sur le premier test à mesurer à savoir le numéro 4 mais divergent sur le reste des tests. Ce résultat montre que ces deux heuristiques ne peuvent donner l'ordre optimal des tests. L'heuristique simple d'une part se base sur la relation entre tests (corrélation) mais ne prend pas en considération les limites des spécifications. L'heuristique de capabilité d'autre part tient compte des limites de spécifications mais néglige la relation entre tests. Il résulte des spécificités de ces deux heuristiques qu'il faut trouver une méthode prenant simultanément en compte la relation entre tests et les limites des spécifications. La méthode d'ordonnancement des tests apporte une solution aux inconvénients des deux heuristiques et fait l'objet de la section suivante.

5.3 La méthode d'ordonnancement des tests

La méthode d'ordonnancement des tests consiste à ordonner les tests de manière à ce que les premiers tests détectent le plus de circuits défectueux. Pour y parvenir, cette méthode se base sur l'estimation du taux de défauts en utilisant la simulation d'un modèle statistique (multinormal, copule ou non paramétrique) construit sur un petit échantillon de données issues de la simulation Monte Carlo ou bien de la production. Une fois le modèle validée, le rééchantillonnage du modèle statistique permet de générer un grand échantillon constitué de plusieurs millions d'instances du circuit sous test. Une estimation précise du taux de défauts, de l'ordre du ppm est ensuite utilisée par les algorithmes de recherche : méthode de séparation et évaluation (branch and bound), algorithmes génétiques et méthodes de recherche flottante (floating search method) pour construire l'ordonnancement des tests.

Dans cette section, nous exposons l'évolution de la méthode d'ordonnancement à partir d'une méthode de réduction de tests fonctionnels utilisant uniquement des données sur les circuits fonctionnels. L'aboutissement de cette évolution est une méthode d'ordonnancement de tests, privilégiant une séquence de tests ou un ordonnancement détectant la plupart des circuits défectueux au plus tôt. Cette méthode propose l'élimination des tests redondants comme une option, qui ne doit être adoptée qu'après plusieurs cycles de test confirmant la non influence des tests éliminés. Un autre atout de cette méthode d'ordonnancement est l'estimation d'intervalles de confiance du taux de défauts à chaque niveau de l'ordonnancement ainsi qu'une adaptativité à la nature des données disponibles. Cette dernière spécificité permet à la méthode d'ordonnancement de proposer un ordre des tests, en se basant sur un petit échantillon de données ne contenant que des circuits fonctionnels, ensuite cet ordre va évoluer au fur et à mesure de la disponibilité de données sur les circuits défectueux qui vont permettre d'utiliser la couverture de fautes catastrophiques pour améliorer l'ordonnancement des tests.

5.3.1 La méthode de réduction de tests fonctionnels

La méthode de réduction de tests fonctionnels permet d'éliminer les performances dont l'impact sur le taux de défauts est minimal. De cette façon, une énumération des tests fonctionnels est obtenue en fonction du taux de défauts. Il est alors possible de choisir le nombre de tests à appliquer en fonction du taux de défauts acceptable. A chaque étape, la méthode élimine les performances une à une et calcule le taux de défauts résultant de chaque élimination. La performance dont la suppression donne un taux de défauts minimal, sera éliminée définitivement de l'ensemble des performances. Cette procédure est répétée jusqu'à ce que l'ensemble des performances soit réduit à une seule performance. Les principales étapes de la méthode sont résumées dans l'organigramme de la Figure 5.2.

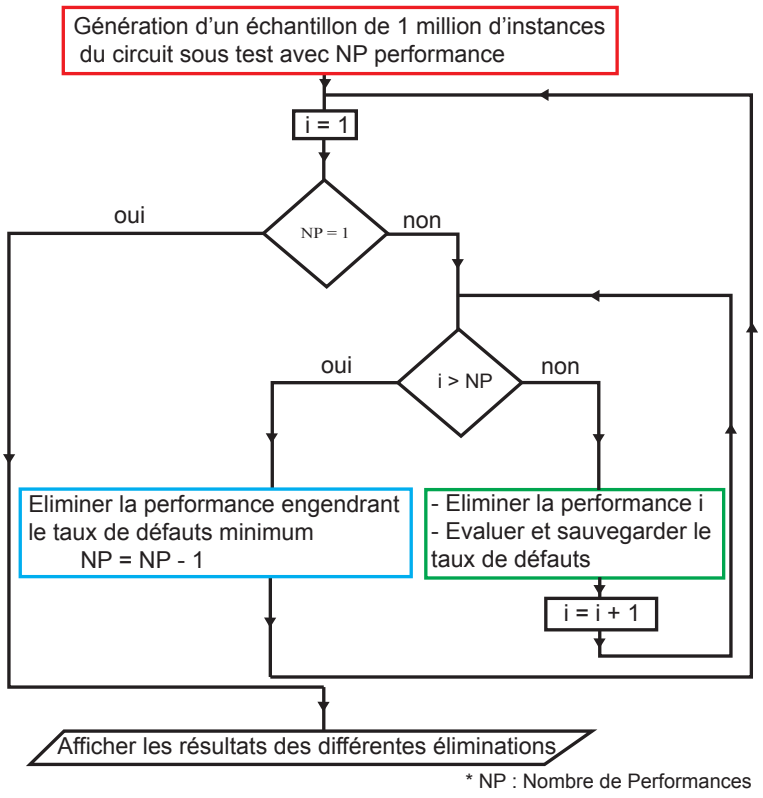


Figure 5.2 – Organigramme de la méthode de réduction de tests fonctionnels.

Pour augmenter la précision des métriques de test, notamment l'estimation du taux de défauts, élément principal de la méthode de réduction de tests fonctionnels, nous avons adopté la technique utilisée dans [17]. Cette technique est appliquée au cas où la loi conjointe des performances et des critères de test suit une loi multinormale, ce qui permet de surmonter la nécessité de disposer d'un grand échantillon de données et de proposer une estimation des métriques de test de l'ordre du ppm.

La simulation Monte Carlo est nécessaire pour obtenir les paramètres de la loi multinormale (vecteur des moyennes et matrice de variance-covariance) des performances du circuit sous test. Puis un rééchantillonnage du modèle statistique permet de générer un échantillon de loi multinormale de grande taille ($\geq$ 1 million).

5.3.1.1 Exemple d'application

L'application de la méthode de réduction de tests fonctionnels à l'amplificateur de la Figure 4.3 avec les spécifications du Tableau 4.3 aura pour objectif d'explorer 3 directions :

- déterminer les performances à éliminer du test pour un certain taux de défauts toléré ;
- établir un ordre d'élimination des performances du test ;
- évaluer le rendement et le taux de défauts pour un ensemble de performances réduit.

La Figure 5.3 présente une trentaine de simulations de la méthode de réduction de tests fonctionnels pour un rendement de 99.99%. L'analyse du graphe montre qu'il est possible de construire un intervalle de confiance de niveau 95% du taux de défauts pour chaque nombre de performances à inclure dans le test. Par exemple pour un taux de défauts variant entre $4ppm$ et $17ppm$ (accepter une erreur de 17 circuits défaillants qui passent le test parmi 1 million de circuits fabriqués), la méthode indique qu'il est possible d'éliminer 4 performances du test. Ainsi on aboutit au test de 7 performances au lieu des 11 performances du départ.

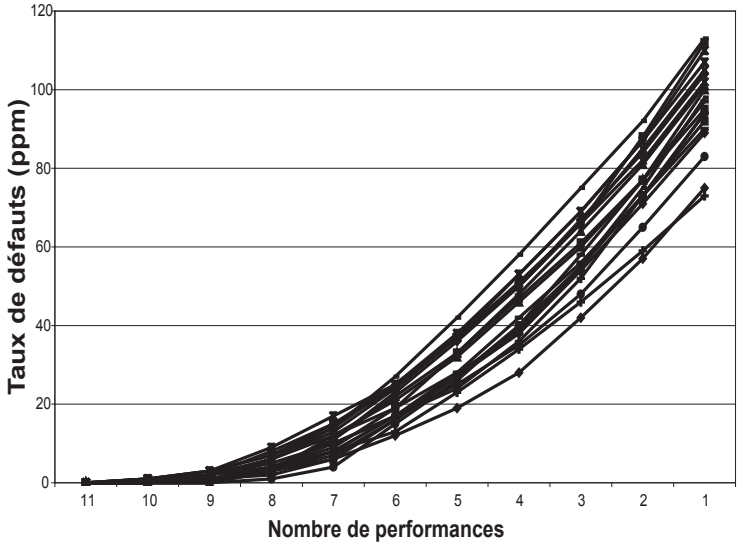


Figure 5.3 – Le taux de défauts en fonction du nombre de performances (30 simulations).

Ce résultat est important, car il permet de spécifier le nombre de performances à appliquer au circuit sous test pour un taux de défauts toléré tout en ayant l'assurance que tous les circuits fonctionnels passent le test. Le Tableau 5.3 résume les intervalles de confiance de niveau 95% du taux de défauts en fonction du nombre de performances dans le test.

Nombre de performances	**Taux de défauts (ppm)**
10	[0 , 1.10001]
9	[0 , 3.0004]
8	[1.0001 , 9.0012]
7	[4.0004 , 17.002]
6	[12.001 , 27.0029]
5	[19.0014 , 42.0039]
4	[28.0018 , 58.0045]
3	[42.0022 , 75.0046]
2	[57.0021 , 92.004]
1	[73.0015 , 113.0026]

Tableau 5.3 – Intervalles de confiance de niveau 95% du taux de défauts.

A partir des résultats de l'application de la méthode de réduction de tests fonctionnels, nous avons observé que ces résultats sont étroitement liés au choix de la marge de tolérance k des spécifications, qui détermine les spécifications de chaque performance ainsi que les fluctuations du processus de simulation. Nous avons appliqué la méthode pour différentes valeurs de cette marge ($k = 1.8, 2.3, 2.8, 3.3, 3.8, 4.3$). Pour chaque valeur, nous avons effectué une trentaine de simulations de la méthode. Pour chacune de ces simulations nous avons comptabilisé le nombre de fois que chaque performance a été éliminée, avec la particularité qu'une performance éliminée au début sera considérée à chaque étape d'élimination. Par contre une performance éliminée à la fin de la méthode ne sera comptée qu'une seule fois. L'ordre d'élimination des performances obtenu est résumé dans le Tableau 5.4.

Ordre d'élimination	**Performance**
1	$SR+$
2	$PSRR\,(G_{ND})$
3	THD
4	$CMRR$
5	GBW_D
6	$PSRR\,(V_{DD})$
7	$Intermodulation$
8	A_D
9	I_{DD}
10	$Noise$
11	$Phase\ Margin$

Tableau 5.4 – Ordre d'élimination des performances.

La comparaison de cet ordre moyen avec l'ordre d'élimination de chacune des simulations $k = 1.8,\ 2.3,\ 2.8,\ 3.3,\ 3.8,\ 4.3$ a montré que cet ordre est le même pour toutes les simulations jusqu'à la $6^{ème}$ performance éliminée et pour les 2 derniers ordre d'élimination ($10^{ème}$ et $11^{ème}$), par contre l'ordre de la $7^{ème}$ jusqu'à la $9^{ème}$ élimination diffère légèrement selon la valeur de la marge de tolérance k des spécifications. L'ordre d'élimination des performances du Tableau 5.4 peut être utilisé pour réduire le temps et le coût du test, car le test des performances les plus influentes va écarter un nombre considérable de circuits défaillants sans avoir recours au test des performances restantes.

5.3.2 Méthode d'ordonnancement des tests

La méthode d'ordonnancement des tests est une amélioration et une généralisation de la méthode de réduction de tests fonctionnels présentée dans la section précédente. Tandis que la méthode de réduction vise l'élimination des tests redondants, nous allons privilégier l'ordonnancement des tests de manière à détecter au plus tôt les circuits défectueux, tout en testant l'ensemble total des tests pour garantir la fiabilité du test. L'étape d'élimination

n'est envisagée que plus tard après validation et confirmation des tests redondants. Donc, en gardant le même principe d'élimination des tests mais en plus optimisé, nous pouvons proposer un ordonnancement optimal des tests avec une estimation des intervalles de confiance sur le taux de défaut à chaque niveau de l'ordonnancement. Les autres atouts de la méthode d'ordonnancement des tests sont les suivants :

1. La modélisation : alors que la méthode précédente ne traitait que les circuits suivant un modèle multinormal, la méthode d'ordonnancement des tests est plus générale car utilisant en plus les copules et la modélisation non paramétrique (voir le chapitre précédent).
2. Ordonnancement : au lieu d'utiliser une heuristique simple pour obtenir l'ordre des tests, on va utiliser une variété d'algorithmes de recherche : branch and bound, algorithmes génétiques et méthodes de recherche flottante. Tous ces algorithmes de recherche feront l'objet de la section 5.4.
3. Rééchantillonnage : la méthode précédente essayait de trouver un ordre d'élimination moyen en considérant chaque génération d'un nouveau million d'instances indépendamment des autres générations, alors qu'on sait que plus la taille des données augmente plus l'ordre obtenu est précis. Donc, on a introduit le principe de cumul des générations d'un million d'instances du circuit sous test. Le cumul signifie qu'à chaque nouvelle génération d'un nouveau million d'instances, on les ajoute aux générations précédentes avant d'ordonner les tests. De cette manière, l'ordre obtenu à chaque itération est plus précis et converge vers un ordre stable. Cet ordre final devient insensible à l'ajout de nouveaux échantillons.
4. Le critère d'arrêt : en plus du critère d'arrêt fixant un nombre prédéfini de générations d'un million d'instances du circuit sous test, on a ajouté un autre critère d'arrêt lié au principe de cumul d'échantillons générés. L'idée du critère d'arrêt supplémentaire est de dire que deux ordres de tests successifs ou plus ne sont pas différents si le taux de défauts de chaque couple de tests de même niveau, ne diffèrent pas plus d'un ppm. Un nombre d'ordres successifs prédéfini, appelé I_{max}, constitue le critère d'arrêt supplémentaire. Par exemple, si on fixe $I_{max} = 5$ cela veut dire que l'exécution de la méthode continue jusqu'à ce que les 5 derniers ordres obtenus ne diffèrent pas de plus de 1 ppm.

Toutes ces améliorations, nous permettent de proposer la méthode d'ordonnancement des tests, dont l'organigramme est représenté sur la Figure 5.4.

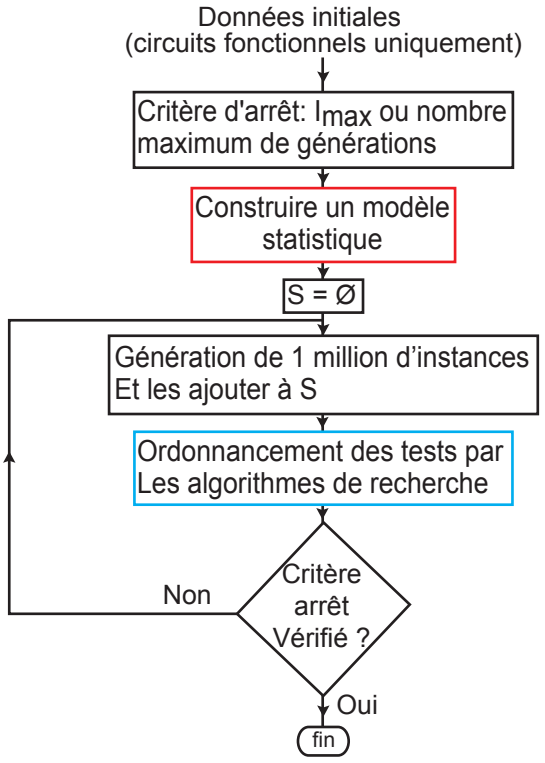


Figure 5.4 – Organigramme de la méthode d'ordonnancement des tests.

L'entrée de la méthode d'ordonnancement des tests demande des données sur les circuits fonctionnels uniquement. Ces données peuvent être obtenues par la simulation Monte Carlo des performances du circuit sous test ou bien des mesures des tests sur des circuits en production. Après avoir spécifier le critère d'arrêt qui peut être un nombre maximum de générations d'un million d'instances du circuit sous test ou bien une valeur du I_{max}, la méthode passe à la modélisation statistique du circuit sous test. Cette modélisation peut aboutir à trois modèles différents : multinormal, copule ou bien non paramétrique. Une fois le modèle statistique validé, l'étape suivante est le cumul des générations d'un million d'instances du circuit sous test et leurs ordonnancements par une des méthodes de recherche : branch and bound, algorithmes génétiques et méthodes de recherche flottante. Cette étape est renouvelée jusqu'à la satisfaction d'un des deux critères d'arrêt (nombre maximum de générations ou bien stabilité des ordres durant les I_{max} dernières itérations). Le résultat final est un ordonnancement des tests détectant au plus tôt les circuits défectueux ainsi qu'un ordre d'élimination des tests avec des intervalles de confiance du taux de défauts correspondant.

5.3.3 Méthode de sélection

Supposons maintenant que nous disposons de données sur des circuits défectueux. Comment ordonner les tests du circuit ? La solution consiste à trouver l'ensemble minimal

des tests couvrant 100% des circuits défectueux. La méthode proposée est constituée de deux étapes :

Etape 1 : cette étape consiste à trouver un ensemble de faible cardinalité C, mais pas forcément minimale, couvrant 100% des circuits défectueux. On démarre avec l'ensemble total des tests S et un ensemble vide de tests retenus $T_{ret} = \emptyset$. Tant qu'on n'a pas traité tous les circuits, on répète les instructions suivantes :

1. trouver le circuit c_i détecté par le minimum de tests $Tmin_i$,
2. trouver le test t_j de $Tmin_i$ qui détecte le maximum de circuits défectueux,
3. éliminer tous les circuits détectés par le test t_j,
4. ajouter le test t_j à l'ensemble des tests retenus ($T_{ret} = T_{ret} \cup \{t_j\}$),
5. enlever le test t_j de l'ensemble des tests S ($S = S - \{t_j\}$),
6. réduire la cardinalité de S ($C = C - 1$).

Etape 2 : elle consiste à explorer toutes les combinaisons de sous-ensemble de cardinalité $C - 1$ parmi les tests de l'ensemble obtenu dans la première étape. Cette dernière étape est répétée jusqu'a l'obtention de l'ensemble minimal et avec une couverture de 100%.

A l'issu de la première étape, on obtient un ensemble avec une couverture de 100% des circuits défectueux, qui va devenir un ensemble minimal à l'issue de la deuxième étape. L'organigramme résumant l'essentiel de ces deux étapes est illustré sur la Figure 5.5.

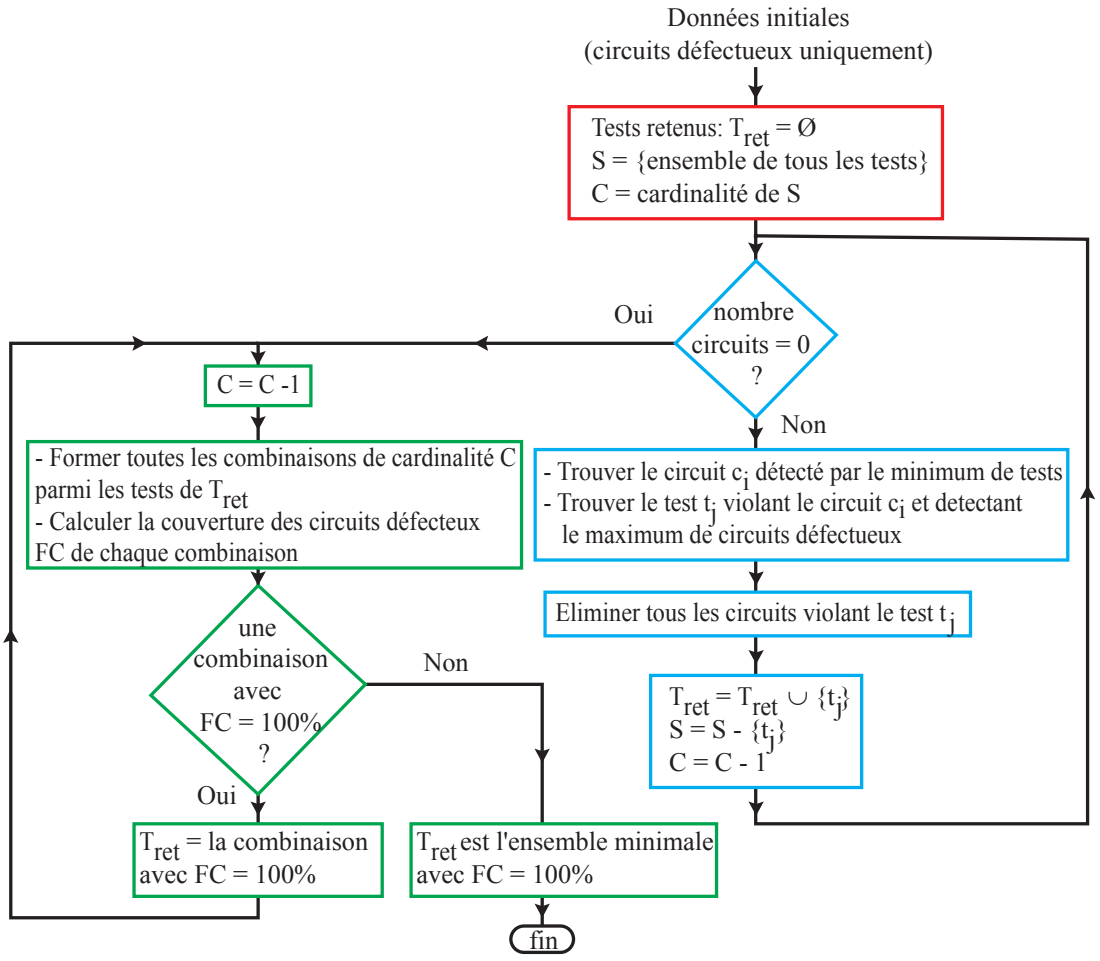


Figure 5.5 – Organigramme de la méthode de sélection.

5.3.4 Méthode de sélection et d'ordonnancement

Comme son non l'indique, cette méthode est un mélange des précédentes méthodes : la méthode d'ordonnancement des tests et la méthode de sélection. Elle est applicable à un circuit pour lequel on a à la fois de données issues de circuits fonctionnels et défectueux ou bien au départ, on ne disposait que de données sur les circuits fonctionnels puis au cours du test, des données sur les circuits défectueux ont pu être récoltées. Le principe de la méthode de sélection et d'ordonnancement est de combiner l'ensemble minimal de tests couvrant 100% des circuits défectueux avec l'ensemble des tests issus de la méthode d'ordonnancement des tests. L'ensemble ainsi formé est ordonnée suivant la valeur décroissante de la couverture des circuits défectueux. Cet ensemble final garantit à la fois la couverture des circuits défectueux ainsi qu'un taux de défauts nul pour les tests non retenus. Les principales étapes de cette méthode sont résumées sur la Figure 5.6.

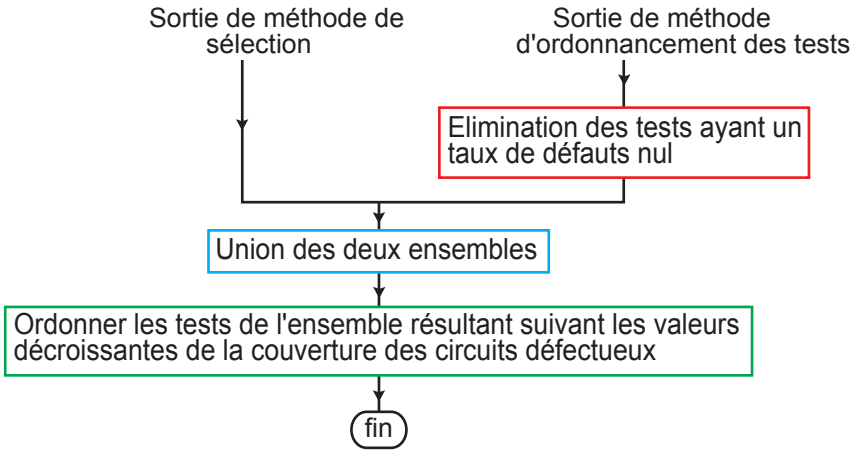


Figure 5.6 – Organigramme de la méthode de sélection et d'ordonnancement.

5.4 Algorithmes de recherche

La sélection de n variables parmi les m possibles ($n << m$) est un problème combinatoire. En effet il existe $n!/(m!(m-n)!)$ sous-ensembles possibles, et le parcours de tous ces sous-ensembles devient impossible avec l'augmentation de la dimension du problème (m très grand). Avec l'augmentation du nombre de tests d'un circuit, le temps d'exécution croit exponentiellement, ce qui rend notre problème NP-complet [63] et difficile à résoudre. Nous allons détailler les différents algorithmes de recherche pour obtenir un ordre d'élimination des tests, en se fixant comme objectif de minimiser la somme des taux de défauts des tests éliminés.

Les algorithmes de recherche peuvent être subdivisés en deux grandes catégories : les algorithmes optimaux garantissant une solution optimale du problème et les algorithmes sous-optimaux (heuristiques). La première catégorie vise à trouver la solution optimale du problème en explorant toutes les solutions possibles, au contraire de la deuxième qui n'a pour objectif que d'approcher la solution optimale.

La seule méthode qui permet l'exploration de tous les cas possibles est la méthode de séparation et évaluation (branch and bound) [64]. Cette méthode exige que le critère d'évaluation soit monotone avec le nombre de variables sélectionnés. Comme cet algorithme n'est applicable que dans le cas de problèmes de petite dimension, on va recourir à des méthodes sous-optimales : les algorithmes génétiques et la recherche flottante (Floating search), pour traiter le cas de circuits avec un nombre important de tests. Ces différents algorithmes seront d'abord détaillés dans les sections suivantes. Par la suite, une autre approche basée sur la décomposition de l'ensemble des tests en sous-ensembles, plus faciles à ordonner par la méthode du branch and bound, sera traitée à la fin du chapitre.

5.4.1 Méthode de séparation et évaluation (branch and bound)

La résolution des problèmes NP-complets est une tâche difficile et requiert des algorithmes efficaces. Le principe du branch and bound est très efficace pour construire une solution à ces problèmes. Un algorithme de séparation et d'évaluation recherche la solution optimale dans l'espace complet des solutions. Toutefois, l'énumération complète de toutes les solutions possibles est difficile et sa complexité croit exponentiellement avec la dimension du problème. L'utilisation conjointe de bornes de la fonction à optimiser ainsi que la valeur de la meilleure solution actuelle permet à l'algorithme de recherche de n'explorer implicitement qu'une partie seulement de l'espace des solutions. A chaque étape de l'algorithme, on dispose d'une meilleure solution et d'un espace inexploré des solutions. L'algorithme commence l'exploration de l'arbre des solutions avec l'ensemble complet des solutions (la racine) et une solution de départ infinie ($-\infty$ pour un problème de maximisation). Chaque itération comporte trois grandes composantes :

1. sélection du nœud à traiter (sélection) ;
2. calcul de la borne du nœud (évaluation) ;
3. séparation de l'ensemble des solutions (nœud) en deux ou plusieurs nœuds (séparation).

La Figure 5.7 illustre le principe de l'algorithme du branch and bound. On représente à gauche l'ensemble des solutions et à droite, un sous-ensemble de solutions sous la forme du sommet d'un arbre. La partie (a) illustre la première étape de la méthode de branch and bound avec l'ensemble complet des solutions S représentant la racine de l'arbre. Puis dans la partie (b) de la figure, on illustre la subdivision de l'ensemble total des solutions S en plusieurs sous-ensembles de solutions : $S1, S2, S3, S4$, cette étape est la phase de séparation de la méthode du branch and bound. Enfin, la partie (c) montre l'évaluation de la borne de chacun des sommets crées précédemment, puis l'arrêt d'exploration des sommets non prometteurs $S1$ et $S4$ (ne contenant pas de solution optimale). Il s'en suit une autre étape de séparation du reste des sommets : $S2$ en $S21$ et $S22$ et le sommet $S3$ en $S23$ et $S24$.

5.4.1.1 Stratégies de séparation

Le parcours d'une arborescence peut être effectué de trois façons :

1. La stratégie «*meilleur d'abord*» BFS [4] : elle consiste à faire des branchements sur le nœud de meilleur solution (borne) jusqu'à atteindre une solution du problème puis de remonter l'arborescence.

4. Best First Search

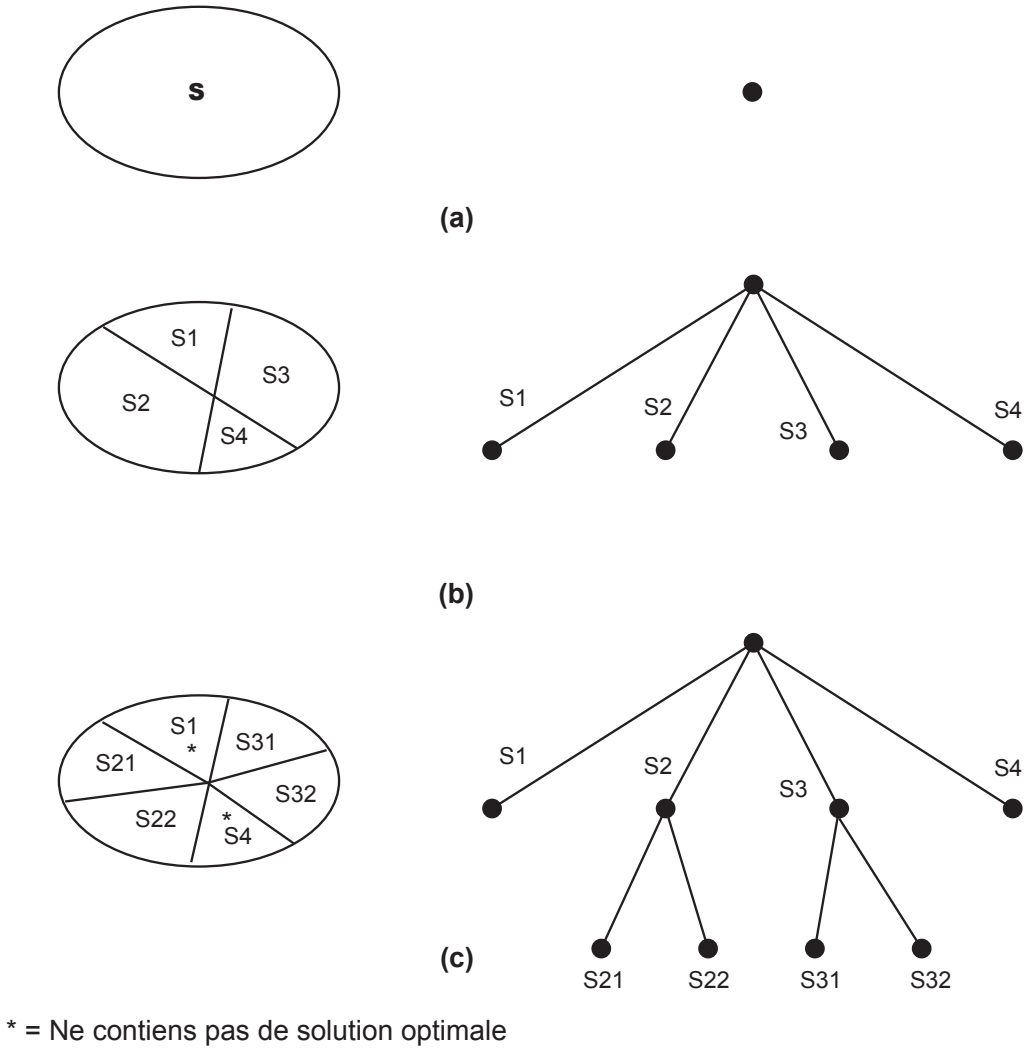


Figure 5.7 – Principe de la Méthode du branch and bound.

2. La stratégie «*profondeur d'abord*» DFS [5] : elle a pour principe de suivre en profondeur un nœud jusqu'à obtenir une solution au problème puis de remonter l'arborescence.
3. La stratégie «*largeur d'abord*» [6] : consiste à explorer en largeur tous les nœuds d'un niveau. Cela revient, dans notre cas, à une énumération complète de l'ensemble des solutions.

Il est clair que ces trois méthodes conduisent à un parcours différent des solutions d'un problème. L'objectif est d'éviter l'énumération de toutes les solutions possibles en s'orientant vers les nœuds les plus prometteurs tout en réduisant le nombre de nœuds explorés.

5. Depth First Search
6. Breadth-First Search

5.4.1.2 Exemple

L'application de la méthode de séparation et évaluation a été réalisée sur l'amplificateur faible bruit représenté sur la Figure 4.9 de la section 4.3.4. La modélisation statistique avec une copule gaussienne a permis de générer un million d'instances du LNA. Cet échantillon va permettre d'ordonner les 5 tests du circuit sous test selon les différentes stratégies de séparation : meilleur d'abord, profondeur d'abord et largeur d'abord. Les résultats d'application sont résumés dans le Tableau 5.5.

N°	**Ordre d'élimination**		
	meilleur d'abord	**profondeur d'abord**	**Largeur d'abord**
1	S_{11}	S_{11}	S_{11}
2	IIP_3	IIP_3	IIP_3
3	NF	NF	NF
4	1-dB CP	1-dB CP	1-dB CP
5	Gain	Gain	Gain

Tableau 5.5 – Ordre d'élimination des tests du LNA par la méthode du branch and bound.

L'ordre d'élimination donnée par les 3 stratégies est le même et constitue l'ordre optimal. Les deux premières approches sont plus rapides car elles n'explorent pas tout l'espace des solutions en coupant tous les chemins non prometteurs. Par contre, la stratégie largeur d'abord explore tous les ordres possibles d'élimination des tests et prend donc plus de temps d'exécution.

5.4.2 Algorithmes génétiques

Les algorithmes génétiques sont des algorithmes évolutionnaires. Les espèces vivantes s'adaptent à leur environnement en évoluant : certains individus se reproduisent pour donner naissance à d'autres individus, d'autres subissent des modifications dans leurs constitutions et certains meurent. Les algorithmes génétiques reproduisent ce modèle d'évolution pour résoudre un problème. Chaque individu représente une solution du problème, qui va évoluer avec les autres solutions (individus) pour atteindre une meilleure solution au problème posé. Avant de détailler le principe des algorithmes génétiques, nous donnons quelques définitions du vocabulaire utilisé, inspiré de la génétique :

- *Individu* : c'est une solution du problème, il est appelé aussi *genotype* ou *chromosome*.
- *Population* : un ensemble d'individus.
- *Gène* : un individu est composé de plusieurs gènes. Dans le codage binaire, un gène correspond à la valeur 0 ou 1.
- *Phénotype* : c'est l'évaluation de la solution ou de l'individu.

L'ensemble de ces définitions est représenté sur la Figure 5.8.

A chaque individu, on associe une évaluation appelé efficacité (adaptation). Cette efficacité est la performance de l'individu dans la résolution du problème posé. Par exemple,

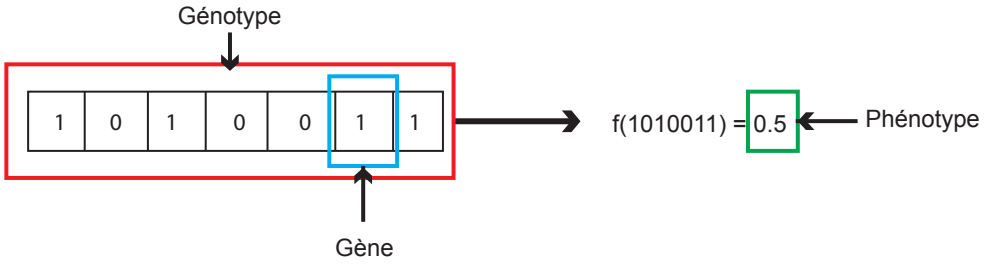


Figure 5.8 – Vocabulaire des algorithmes génétiques.

dans un problème de maximisation d'une fonction f (profits, distance, ...), l'efficacité augmente avec l'aptitude de l'individu à faire croître cette fonction.

Pour une population de N individus, l'efficacité de l'individu i est calculée suivant la formule suivante :

$$\text{efficacite}(i) = \frac{f(i)}{\sum_{j=1}^{N} f(j)} N \; \forall i = 1, \ldots, N. \tag{5.8}$$

Après le calcul de l'efficacité des individus d'une population, on opère une reproduction. Cette dernière est la combinaison du processus d'évaluation et de sélection. Elle permet la copie d'un individu d'une génération à une autre. La sélection des individus est réalisée par le système de roulette[7]. Ce système associe à chaque individu un secteur de taille proportionnelle à son efficacité. Ainsi, les meilleurs individus (plus grande efficacité) ont plus de chance d'être sélectionnés et de participer à l'amélioration de la population des solutions. Ce mécanisme est illustré sur la Figure 5.9.

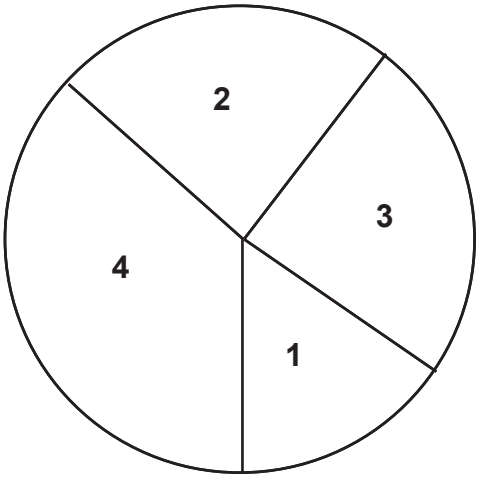


Figure 5.9 – Schéma d'une roulette.

Une fois les individus sélectionnés, ils vont subir des opérations de croisement et de mutation pour former de nouveaux individus. Le croisement simule la reproduction d'individus pour en créer de nouveaux. Pour cela, on sélectionne aléatoirement un point de

7. Roulette Wheel Selection

croisement dans le codage de l'individu. En ce point, on sépare le codage et on échange les parties de droite pour former deux nouveaux individus. Une illustration de ce principe est montrée sur la Figure 5.10. La mutation d'un gène au niveau d'un individu consiste

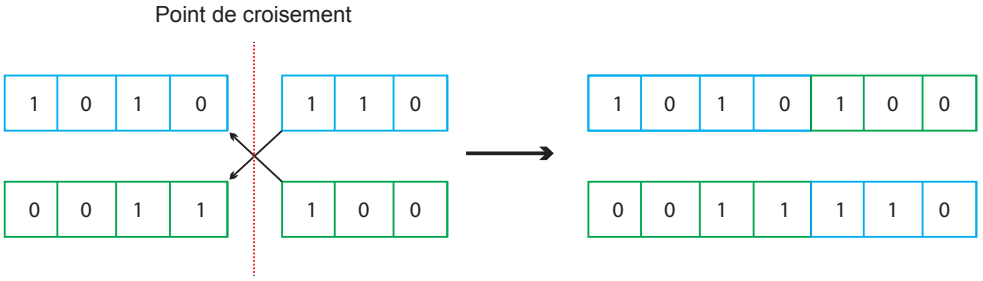


Figure 5.10 – Le croisement.

à choisir aléatoirement un de ses gènes et changer sa valeur. Généralement, la proportion des individus à muter est très faible. Ce principe est illustré sur la Figure 5.11.

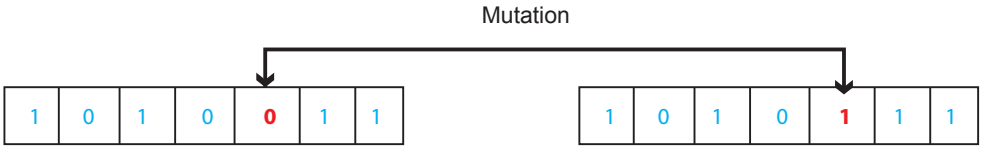


Figure 5.11 – La mutation.

5.4.2.1 Exemple

L'application de l'algorithme génétique a été réalisée sur les mêmes données de l'amplificateur faible bruit de la section précédente. Chaque individu représente un ordre d'élimination des tests avec 5 gènes et l'évaluation d'un individu est représentée par la somme des taux de défauts liés aux éliminations des tests. Sur une population initiale de 20 individus générés aléatoirement, on applique un algorithme génétique avec une probabilité de croisement de 0.75 et de mutation de 0.1. Après un maximum de 50 générations, on retrouve le même résultat optimal obtenu précédemment par la méthode du branch and bound, et résumé dans le Tableau 5.6.

N°	Ordre d'élimination
1	S_{11}
2	IIP_3
3	NF
4	1-dB CP
5	Gain

Tableau 5.6 – Ordre d'élimination des tests du LNA par un algorithme génétique.

Le nombre restreint de tests du LNA a permis à l'algorithme génétique de retrouver l'ordre d'élimination des tests optimal. Ceci démontre l'efficacité de cette méthode

heuristique dans un cas d'étude avec un petit nombre de tests et un choix judicieux des paramètres de l'algorithme génétique. Avec un autre circuit comportant un nombre important de tests, il faudra varier les paramètres de l'algorithme génétique et faire plusieurs expérimentations pour aboutir à une solution proche de l'optimale. Cette méthode est une alternative à la méthode du branch and bound quand le nombre de tests est important et que la méthode optimale de branch and bound devient inapplicable.

5.4.3 Méthode de recherche flottante (Floating Search)

L'algorithme du branch and bound n'est pas applicable dans tous les problèmes d'optimisation, car il a une complexité exponentielle avec l'augmentation de la dimension du problème et exige que le critère d'évaluation soit monotone. Pour pallier ces limitations, on fait appel à des heuristiques basées sur le parcours séquentiel de l'ensemble des solutions. Ces méthodes de sélection de variables se basent sur l'ajout et l'élimination itérative de variables. Il existe deux approches de parcours : démarrer à partir d'un ensemble vide puis ajouter au fur et à mesure des variables (SFS [8]) ou partir de l'ensemble complet des variables puis éliminer des variables sélectionnées (SBS [9]).

Les deux algorithmes SFS et SBS présentent l'avantage d'être faciles à implémenter et d'avoir un temps d'exécution court. Toutefois, l'inconvénient majeur de ce type de méthodes est de ne pas pouvoir remettre en cause un choix déjà validé : l'ajout d'une variable dans SFS et l'élimination d'une variable dans SBS. Pour remédier à ces inconvénients, il existe des méthodes qui combinent les deux algorithmes SFS et SBS, en permettant d'ajouter et d'éliminer des variables. L'algorithme "plus l take away r" [65] est une illustration de la combinaison de SFS et SBS. Son principe est de répéter l fois la fonction SFS puis r fois la fonction SBS. Le choix des ces deux paramètres est crucial pour la convergence de la méthode et sa vitesse d'exécution.

La méthode "plus l take away r" a été généralisée pour construire des méthodes flottantes (Floating). Ces méthodes permettent de s'affranchir de la valeur statique des deux paramètres l et r. Dans la version SFFS (Sequential Floating Forward Selection) dont l'algorithme est représenté sur la Figure 5.1, on exécute, après chaque étape de forward, des étapes backward tant qu'un critère d'évaluation est amélioré. Ces deux étapes sont alternées jusqu'à la satisfaction d'un critère d'arrêt. Dans la version SBFS, le même principe est appliqué avec inversion des deux étapes. Les deux méthodes SFFS et SBFS ont été généralisées, en rendant le nombre d'étapes de forward et de backward variables et dépendant d'un critère d'évaluation pour former les méthodes adaptatives : Adaptive SFFS (ASFFS) et Adaptive SBFS (ASBFS).

8. Sequential Forward Selection
9. Sequential Backward Selection

Entrées : $V = \{x_j | j = 1, \ldots, p\}$
/* V : ensemble des variables initiales
J : critère (à maximiser par exemple) */

Initialisation :
$q \leftarrow 0$
$F_q \leftarrow \Phi$

Etape 1 (forward)
$x_+ \leftarrow argmax_{x_j \in V \setminus F_q} J(F_q \cup \{x_j\})$
$F_{q+1} \leftarrow F_q \cup \{x_+\}$
$q \leftarrow q + 1$

Etape 2 (backward)
$x_- \leftarrow argmax_{y_j \in F_q} J(F_q \setminus \{y_j\})$
si $J(F_q \setminus \{x_-\}) > J(F_{q-1})$ **alors**
  $F_{q-1} \leftarrow F_q \setminus \{x_-\}$
  $q \leftarrow q - 1$
  Aller à Etape 2
sinon
  Aller à Etape 1
fin si
Sorties : F_q : sous-ensemble de q variables sélectionnées

ALGORITHME 5.1 – L'algorithme SFFS.

5.4.3.1 Exemple

Comme pour les deux méthodes précédentes, l'application des méthodes de recherche flottante à été réalisée sur les données simulées avec une copule gaussienne sur le LNA. Les méthodes expérimentées sont : SFFS, SFBS, ASFFS et ASFBS et les résultats d'application sont résumés dans le Tableau 5.7.

N°	**Ordre d'élimination**			
	SFFS	**SFBS**	**ASFFS**	**ASFBS**
1	S_{11}	S_{11}	S_{11}	S_{11}
2	IIP_3	IIP_3	IIP_3	IIP_3
3	NF	NF	NF	NF
4	1-dB CP	1-dB CP	1-dB CP	1-dB CP
5	Gain	Gain	Gain	Gain

Tableau 5.7 – Ordre d'élimination des tests du LNA par les méthodes de recherche flottante.

Ces méthodes de recherche flottante ont permis de retrouver l'ordre optimal d'élimination des tests car la dimension du problème est petite. Elles constituent avec les algorithmes génétiques une alternative pour approcher la solution optimale dans le cas où le nombre de tests devient important et non résoluble par la méthode exacte du branch and bound.

5.5 Méthode de décomposition

Ordonner les tests d'un circuit sous test peut être réalisé par la méthode optimale du branch and bound, dans le cas où le nombre de tests est réduit, ou dans le cas contraire avec des méthodes sub-optimales : algorithmes génétiques et méthodes de recherche flottante. La méthode proposée dans cette section décompose l'ensemble des tests en sous-ensembles de faible cardinalité pour pouvoir les ordonner d'une manière optimale par la méthode du branch and bound. La combinaison des ordres des sous-ensembles permet de construire l'ordre final des tests. Cette démarche a pour avantage l'utilisation d'une méthode optimale au lieu d'une approche heuristique et tout particulièrement dans le cas d'un circuit modélisé par une loi non paramétrique. Dans ce dernier cas, la construction d'un modèle robuste exige de disposer d'un échantillon de départ (Monte Carlo ou production) dont la taille minimale croit exponentiellement avec le nombre de tests du circuit sous test. Le fait de travailler avec des sous-ensembles de petite taille améliore la modélisation et diminue le temps d'ordonnancement.

5.5.1 Principe de la méthode de décomposition

Les principales étapes de la méthode de décomposition sont illustrées dans l'organigramme de la Figure 5.12.

Les deux paramètres de base de l'algorithme sont la cardinalité c des sous-ensembles et le nombre e de tests remplacés dans un sous-ensemble pour en former un nouveau. Un sous-ensemble initial de c tests est choisi aléatoirement parmi l'ensemble des tests S_n. Ce sous-ensemble est modélisé, puis rééchantillonné et enfin ordonné d'une manière optimale avec la méthode du branch and bound (section 5.3.2). Les derniers e tests de l'ordonnancement, ayant un fort impact sur le taux de défauts, sont éliminés et remplacés par e tests non encore traités. Cette étape est répétée jusqu'à ce que tous les tests de S_n soient traités. Alors, le premier test du sous-ensemble final est définitivement éliminé de l'ensemble des tests S_n. Il correspond au test le moins important en termes de détection de circuits défectueux. L'algorithme est répété jusqu'à ce que l'ensemble des tests S_n soit vide. Le choix des deux paramètres c et e est important pour la convergence de la méthode et le temps d'exécution.

5.5.2 Application

L'application a été réalisée sur deux cas d'études. Le premier est un circuit artificiel avec un nombre important de tests et simulé suivant un modèle statistique gaussien. Le deuxième cas d'étude est l'amplificateur opérationnel présenté dans la section 4.2.2.3 du chapitre précédent.

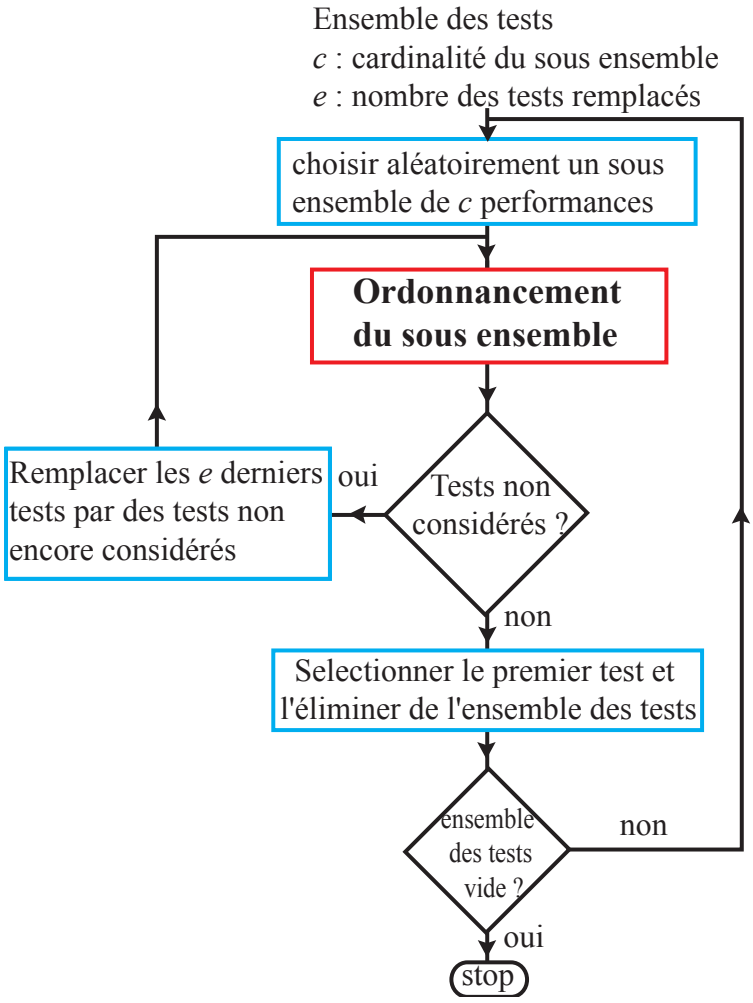


Figure 5.12 – Organigramme de la méthode de décomposition.

5.5.2.1 Circuit artificiel

Un circuit artificiel a été simulé pour valider la méthode d'ordonnancement basée sur la décomposition de l'ensemble des tests en sous-ensembles de petite cardinalité. On a construit un circuit artificiel incluant 20 tests $(X_1, \ldots, X_{20})$ suivant un modèle multinormal avec des paramètres aléatoires (vecteur des moyennes et matrice de variance-covariance). Les spécifications de ces tests varient entre 3σ et 4σ, avec $\mu_1 \pm 3\sigma_1$ pour le test X_1, $\mu_{20} \pm 4\sigma_{20}$ pour le test X_{20}, avec une incrémentation régulière entre les deux limites 3σ et 4σ pour les autres tests. Si les tests sont indépendants, le test X_{20} sera le premier test à être éliminé car possédant des limites de test plus larges et donc plus difficile à violer. Puis viendra le test X_{19} et ainsi de suite jusqu'au test X_1 $(X_{20}, X_{19}, X_{18}, \ldots, X_1)$. Toutefois, à cause des corrélations aléatoires entre les tests, l'ordre exact des tests sera différent.

Pour ce circuit artificiel avec 20 tests suivant un modèle multinormal, il est possible de considérer un seul ensemble comprenant les 20 tests. On exécute donc l'algorithme de

la la Figure 5.4 en utilisant le modèle multinormal pour obtenir l'ordre exact des tests qu'on va appeler l'ordre multinormal de référence dans le Tableau 5.8.

On peut observer dans la colonne de l'ordre de référence que le premier test à éliminer est le test X_{19}, avec un taux de défaut de $D_1 = 37.6$ ppm. Le deuxième test à éliminer est le test X_{18}, qui ajoute 46.5 ppm au taux de défaut pour atteindre $D_2 = 84.1$ ppm et ainsi de suite jusqu'au dernier qui indique que ce circuit artificiel possède une population de 13842 circuits défectueux. Pour obtenir cet ordre de référence, l'algorithme de la Figure 5.4 a requis 57 itérations et un temps de simulation de 200 secondes sur une machine Pentium-4 avec un processeur de 3 GHz et 1 Go de mémoire.

On a considéré ensuite l'algorithme de décomposition de la Figure 5.4 avec des sous-ensembles suivant un modèle multinormal et un modèle non paramétrique (KDE). Tous ces résultats sont comparés à l'ordre obtenu par l'heuristique simple basée sur la corrélation (section 5.2.1). L'ensemble des résultats est résumé dans le Tableau 5.8.

Ordre elimination	**Reference multinormal**		**sous-ensemble multinormal**			**sous-ensemble KDE**			**Heuristique corrélation**		
i	**ordre**	D_i	**ordre**	D_i	ΔD_i	**ordre**	D_i	ΔD_i	**ordre**	D_i	ΔD_i
1	19	37.6	20	53.4	15.8	18	46.1	8.5	11	163.9	126.3
2	18	84.1	19	91.1	7	17	111	26.9	14	275.6	191.6
3	20	137.8	18	137.8	**0**	20	164.9	27.2	16	339.5	201.8
4	16	193.7	17	203.3	9.6	16	221.1	27.4	18	387.8	194.1
5	17	259.5	16	259.5	**0**	19	259.5	**0**	7	1045.1	785.6
6	14	372.4	15	420.9	48.5	15	420.9	48.5	17	1116.7	744.4
7	15	534.5	14	534.5	**0**	14	534.5	**0**	8	1604.6	1070.2
8	11	717.8	13	753.4	35.6	13	753.4	35.6	19	1647.5	929.7
9	13	937.1	11	937.1	**0**	10	1184.7	247.6	6	2532	1594.8
10	12	1246	12	1246	**0**	12	1496.5	249.7	2	4563.2	3316.4
11	10	1681.4	10	1681.4	**0**	11	1681.4	**0**	13	4808.7	3127.3
12	9	2152.8	9	2152.8	**0**	9	2152.8	**0**	12	5123.7	2971
13	8	2639	8	2639.2	**0**	7	2866.4	227.2	4	6633	3993.8
14	7	3366	7	3366	**0**	8	3366	**0**	20	6693.3	3327.3
15	6	4299	6	4299	**0**	4	4831.7	532.7	3	8555.7	4256.7
16	5	5587.2	5	5587.2	**0**	6	5795.9	208.6	15	8737.8	3150.5
17	4	7090.8	4	7090.8	**0**	5	7090.8	**0**	10	9243.7	2152.9
18	3	8930.6	3	8930.6	**0**	1	9758.2	827.6	1	11917.6	2987
19	2	11169.9	2	11169.9	**0**	3	11598.3	428.3	5	13225.8	2055.9
20	1	13842	1	13842	0	2	13842	0	9	13842	0

Tableau 5.8 – Ordonnancement des tests du circuit artificiel par la méthode de décomposition.

L'analyse des résultats du Tableau 5.8 montre que l'heuristique simple basée sur la corrélation produit un ordre complètement différent de celui de la référence. Par contre la méthode de décomposition produit un bon ordre d'élimination des tests, proche de la référence, avec de meilleurs résultats pour le modèle multinormal que pour le modèle non paramétrique. Ceci peut s'expliquer par le fait que le circuit a été généré suivant un modèle multinormal.

5.5.2.2 Amplificateur opérationnel

Le deuxième cas d'étude est l'amplificateur opérationnel décrit dans la section 4.2.2.3 du chapitre précédent. On commence par obtenir l'ordre de référence multinormal quand on considère l'ensemble total des tests. Le Tableau 5.9 résume l'application de la méthode de décomposition avec un modèle multinormal et non paramétrique ainsi que l'application de l'heuristique simple basée sur la corrélation.

Ordre elimination	**Reference multinormal**		**sous-ensemble multinormal**			**sous-ensemble KDE**			**Heuristique corrélation**		
i	**ordre**	D_i	**ordre**	D_i	ΔD_i	**ordre**	D_i	ΔD_i	**ordre**	D_i	ΔD_i
1	5	0.4	5	0.4	**0**	7	4.7	4.3	10	2	1.6
2	2	2.4	2	2.4	**0**	2	6.7	4.3	7	6.8	4.3
3	4	5	4	5	**0**	10	15.3	10.3	6	7.7	2.7
4	7	9.8	7	9.8	**0**	11	23.6	13.8	2	16.2	6.5
5	6	16.8	9	17.2	0.4	3	32	15.2	4	18.4	1.6
6	9	24.4	6	24.4	**0**	9	39.3	15	9	25.9	1.5
7	11	32.7	10	33	0.2	1	47.8	15	8	33.2	0.5
8	3	41.1	1	41.5	0.4	6	48.8	7.7	1	41.7	0.6
9	1	49.6	11	49.8	0.2	5	52.2	2.7	11	50.1	0.5
10	8	58.1	3	58.2	**0**	8	58.4	0.3	3	58.4	0.3
11	10	66.7	8	66.7	0	4	66.7	0	5	66.7	0

Tableau 5.9 – Ordonnancement des tests de l'amplificateur opérationnel par la méthode de décomposition.

L'analyse des résultats du Tableau 5.9 montre une fois de plus que l'ordre obtenu avec la décomposition multinormale est plus proche de la référence que celui obtenu par le modèle non paramétrique ainsi que de l'ordre obtenu par l'heuristique simple basée sur la corrélation.

Dans le but de voir l'effet des deux paramètres c et e sur la méthode de décomposition, on considère la somme des erreurs commises après chaque élimination d'un test en utilisant le modèle multinormal. Le Tableau 5.10 montre l'influence de ces deux paramètres, en utilisant le même critère d'arrêt $I_{max} = 10$ et pour différentes valeurs de c et e. Ces résultats sont illustrés sur la Figure 5.13.

e	$\mathbf{I_{max}}$	$\Sigma\Delta D_i$					
		c = 5	**c = 6**	**c = 7**	**c = 8**	**c = 9**	**c = 10**
1	10	21.5	9.1	6.1	1.2	1.5	3.4
2	10	17.5	17.9	3.1	3.9	1.3	2.6
3	10	20.2	10.8	7.9	9.1	1.4	2.3
4	10	33.5	10.8	6.8	3.6	5.1	2.9

Tableau 5.10 – Somme des erreurs de la méthode de décomposition pour différent paramètres.

L'analyse de la Figure 5.13 montre une réduction de l'erreur commise avec l'augmentation de la cardinalité c des sous-ensembles. En effet, avec des sous-ensembles de plus grande cardinalité, on conserve plus d'information sur les tests. On remarque aussi une diminution des erreurs quand la valeur de e décroît. Ceci s'explique par le fait qu'avec de

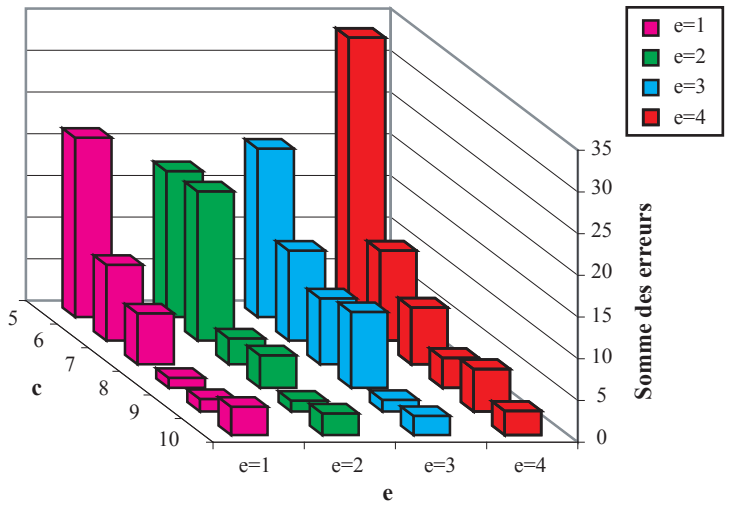


Figure 5.13 – Influence de la cardinalité des sous-ensembles c et du nombre de tests remplacés par sous-ensemble e sur la somme des erreurs de la méthode d'ordonnancement des tests.

petites valeur du paramètre e, le nombre de sous-ensembles traité sera plus important et donc des résultats plus précis.

Pour ce cas d'étude, nous avons validé l'ordre des tests dans le cas de circuits avec fautes catastrophiques. On considère des fautes catastrophiques résultantes de circuit ouvert ou fermé sur toutes les résistances et condensateurs. Le modèle de fautes est constitué de 160 fautes catastrophiques. Le Tableau 5.11 montre l'évolution de la couverture de fautes catastrophiques suivant l'ordre d'élimination des tests obtenu avec une décomposition en sous-ensembles de loi multinormale ainsi que l'ordre de référence.

Ordre d'élimination	**Référence multinormal**		**Sous-ensemble multinormal**	
i	**ordre**	**FC**(%)	**ordre**	**FC**(%)
0	tous les tests	**98.13**	tous les tests	**98.13**
1	$PSRR\ (G_{ND})$	**98.13**	$PSRR\ (G_{ND})$	**98.13**
2	GBW_D	**98.13**	GBW_D	**98.13**
3	$CMRR$	**98.13**	$CMRR$	**98.13**
4	THD	**98.13**	THD	**98.13**
5	$PSRR\ (V_{DD})$	**98.13**	$Intermodulation$	97.5
6	$Intermodulation$	97.5	$PSRR\ (V_{DD})$	97.5
7	$Noise$	97.5	SR	97.5
8	$Phase\ Margin$	97.5	A_D	97.5
9	A_D	97.5	$Noise$	97.5
10	I_{DD}	93.75	$Phase\ Margin$	95.63
11	SR	0	I_{DD}	0

Tableau 5.11 – La couverture de fautes catastrophiques de l'amplificateur opérationnel.

Une couverture de fautes catastrophiques (FC) maximale de 98.13% est obtenue quand tous les tests sont vérifiés. L'évolution de la couverture de fautes catastrophiques diffère légèrement entre l'ordre de référence et l'ordre de décomposition. Toutefois, dans les deux cas, la couverture de fautes catastrophiques maximale est obtenue durant l'élimination de

4 tests sur les 11 tests du départ. Ceci montre que la méthode de décomposition proposée conserve une bonne couverture de fautes catastrophiques suivant l'élimination des tests.

5.6 Application

L'application de la méthode d'ordonnancement des tests a été réalisée sur l'amplificateur opérationnel (Figure 4.3) et le LNA (Figure 4.9). L'amplificateur opérationnel est modélisé avec une loi multinormale et ordonné par la méthode du branch and bound. Quand au LNA, la modélisation a été faite avec une copule gaussienne et l'ordre établi avec la méthode du branch and bound.

L'application des deux autres méthodes, la méthode de sélection et la méthode de sélection et d'ordonnancement, n'a pu être réalisé sur l'amplificateur opérationnel et le LNA. Ceci est dû à la nature des données disponible pour les deux circuits. Il s'agit de simulation Monte Carlo ne contenant aucun circuit défectueux. Toutefois, ces méthodes seront appliquées sur le circuit industriel du chapitre suivant car il comporte des données de production avec des circuits défectueux.

5.6.1 Amplificateur opérationnel

Le résultat de l'application de la méthode d'ordonnancement des tests sur l'amplificateur est un ordre de tests détectant au plus tôt les circuits défectueux, donné dans le Tableau 5.12.

Ordre	Test
1	$SR+$
2	I_{DD}
3	A_D
4	$Phase\ Margin$
5	$Noise$
6	$Intermodulation$
7	$PSRR\,(V_{DD})$
8	THD
9	$CMRR$
10	GBW_D
11	$PSRR\,(G_{ND})$

Tableau 5.12 – Ordonnancement des tests de l'amplificateur opérationnel.

Cet ordre montre qu'il faudra tester en premier le test $SR+$ puis le test I_{DD} et ainsi de suite jusqu'au dernier test $PSRR\,(G_{ND})$. Cette séquence de tests assure la détection des circuits défectueux au plus tôt. Un autre résultat de l'application est la construction d'intervalles de confiance à 95% du taux de défauts (Tableau 5.13).

Ordre d'élimination	**Test**	**Taux de défauts à** 95%
1	$PSRR\,(G_{ND})$	[0.48 , 0.49]
2	GBW_D	[2.49 , 2.51]
3	$CMRR$	[4.99 , 5.02]
4	THD	[9.78 , 9.81]
5	$PSRR\,(V_{DD})$	[16.76 , 16.82]
6	$Intermodulation$	[24.19 , 24.29]
7	$Noise$	[32.51 , 32.65]
8	$Phase\ Margin$	[41.06 , 41.21]
9	A_D	[49.72 , 49.90]
10	I_{DD}	[58.50 , 58.71]
11	$SR+$	[67.40 , 67.67]

Tableau 5.13 – Intervalles de confiance de l'amplificateur opérationnel au niveau 95% du taux de défauts.

L'analyse du Tableau 5.13 montre que l'élimination du premier test $PSRR\,(G_{ND})$, va engendrer un taux de défaut compris dans l'intervalle [0.48 , 0.49]. Les intervalles de confiance du taux de défauts permettent de déterminer l'ensemble des tests qu'on peut éliminer, suivant le taux de défaut toléré.

L'injection de 160 fautes catastrophiques (court-circuit, circuit ouvert) dans l'amplificateur a permis de calculer la couverture de fautes catastrophiques. Le calcul de cette métrique suivant l'ordre d'élimination des tests du Tableau 5.13 est illustré dans le Tableau 5.14.

Ordre d'élimination	**Test**	**Couverture de fautes catastrophiques** (%)
0	tous les tests	**98.13**
1	$PSRR\,(G_{ND})$	**98.13**
2	GBW_D	**98.13**
3	$CMRR$	**98.13**
4	THD	**98.13**
5	$PSRR\,(V_{DD})$	**98.13**
6	$Intermodulation$	97.5
7	$Noise$	97.5
8	$Phase\ Margin$	97.5
9	A_D	97.5
10	I_{DD}	95.63
11	SR	0

Tableau 5.14 – La couverture de fautes catastrophiques de l'amplificateur opérationnel.

Du Tableau 5.14, on remarque que la couverture de fautes catastrophiques est de 98.13% quand tous les tests sont exécutés. Cette couverture maximale reste la même pendant l'élimination des 5 premiers tests. Puis, sa valeur décroît avec l'élimination des autres tests. Ce résultat montre qu'avec un ordre d'élimination établi uniquement sur la base des fautes paramétriques, cet ordre reste valide dans le cas des fautes catastrophiques. Dans ce cas, on recommande de tenir compte des intervalles de confiance du Tableau 5.13 dans la décision d'élimination des tests.

5.6.2 LNA

Le premier résultat de l'application de la méthode d'ordonnancement des tests sur le LNA est un ordonnancement des test, résumé dans le Tableau 5.15.

Ordre	**Test**
1	Gain
2	S_{11}
3	IIP_3
4	NF
5	1-dB CP

Tableau 5.15 – Ordonnancement des tests du LNA.

La séquence de test : Gain, S_{11}, IIP_3, NF et 1-dB CP est le meilleure ordre pour détecter les circuits défectueux au plus tôt. Le deuxième résultat de l'application de la méthode d'ordonnancement des tests est la construction d'intervalles de confiance à 95% du taux de défauts (Tableau 5.16).

Ordre d'élimination	**Test**	**Taux de défauts à** 95%
1	1-dB CP	[0.4 , 0.45]
2	NF	[34.19 , 34.91]
3	IIP_3	[69.39 , 70.35]
4	S_{11}	[111.73 , 112.53]
5	Gain	[1091.49 , 1096.78]

Tableau 5.16 – Intervalles de confiance du LNA au niveau 95% du taux de défauts.

L'élimination du premier test 1-dB CP va engendrer un taux de défaut compris dans l'intervalle [0.4 , 0.45]. Avec la deuxième élimination du test NF, le taux de défaut passe à [34.19 , 34.91]. Suivant le taux de défaut toléré, ces intervalles de confiance définiront l'ensemble des tests à éliminer.

L'injection d'un modèle de fautes composé de 50 fautes catastrophiques (court-circuit, circuit ouvert) dans le LNA permet le calcul de la couverture de fautes catastrophiques. En suivant l'ordre d'élimination des tests du Tableau 5.16, on obtient les couvertures de fautes catastrophiques du Tableau 5.17.

Ordre d'élimination	**Test**	**Couverture de fautes catastrophiques** (%)
0	tous les tests	**86**
1	1-dB CP	82
2	NF	78
3	S_{11}	78
4	THD	74
5	Gain	0

Tableau 5.17 – La couverture de fautes catastrophiques du LNA.

L'application de tous les tests engendre une couverture de fautes catastrophiques maximale de 86%. Cette couverture est dégradée dès la première élimination du test 1-dB CP pour atteindre une couverture de 82%. Dans ce cas, la désignation de l'ensemble des tests à éliminer doit être un compromis entre le taux de défaut et la couverture de fautes catastrophiques toléré.

5.7 Conclusion

Dans ce chapitre, nous avons présenté les différentes méthodes utilisées pour l'ordonnancement des tests d'un circuit sous test. Après la modélisation et le rééchantillonnage du modèle, on dispose d'un grand échantillon de plusieurs millions d'instances. L'ordonnancement des tests est fait sur la base de l'estimation du taux de défauts à chaque élimination d'un test de l'ensemble des tests. Si la cardinalité du circuit sous test est petite, alors l'utilisation de la méthode optimale du branch and bound fournit l'ordonnancement optimal des tests. Toutefois, avec des circuits complexes, ayant plusieurs centaines de tests, la méthode du branch and bound devient inefficace et consomme trop de temps d'exécution. On fait alors appel aux heuristiques, algorithmes génétiques et recherche flottante, pour ordonner les tests du circuit.

La méthode d'ordonnancement des tests s'adapte à la nature des données disponibles. Si on ne dispose que de données sur les circuits fonctionnels, alors la méthode d'ordonnancement des tests permet d'ordonner les tests de manière à détecter au plus tôt le plus de circuits défectueux et ainsi réduire le temps et le coût du test. Un autre résultat de la méthode d'ordonnancement est de proposer des intervalles de confiance sur les taux de défauts à chaque niveau d'élimination d'un test. Ainsi, elle propose une estimation du risque d'erreur lié à chaque élimination de test. Cette méthode a été appliquée avec succès sur un amplificateur opérationnel et un LNA. L'ordre d'élimination des tests a été validé par l'utilisation d'un modèle de fautes et le calcul de la couverture de fautes catastrophiques.

Nous avons ensuite proposé la méthode de sélection qui traite les données sur les circuits défectueux. Cette méthode aboutit à un ensemble minimal de tests couvrant 100% des circuits défectueux. Enfin, si l'on dispose à la fois de données sur les circuits fonctionnels et défectueux, alors la méthode de sélection et d'ordonnancement permet de combiner le résultat des deux précédentes méthodes pour proposer un ensemble de tests assurant en même temps une couverture de 100% des circuits défectueux et un taux de défauts paramétrique nul.

Une autre approche d'ordonnancement de tests consiste à décomposer l'ensemble des tests en sous-ensembles plus faciles à modéliser et à ordonner. Cette méthode est bien adaptée aux circuits complexes, avec un grand nombre de tests, et particulièrement dans le cas d'une modélisation non paramétrique exigeant un échantillon de départ de grand taille.

Chapitre 6

Résultats expérimentaux

6.1 Introduction

Dans ce chapitre, nous allons appliquer la méthode d'ordonnancement des tests sur un circuit industriel. Jusqu'à présent, les méthodes que nous avons développées ont été appliquées avec succès sur des données issues de la simulation Monte Carlo. Elles ont été validées dans le cas d'un amplificateur opérationnel avec l'utilisation du modèle multi-normal. Ainsi que sur un LNA avec le modèle non paramétrique. Dans les cas d'études industriels, la dimension du problème devient plus complexe car le nombre de tests est très important. Notre choix s'est porté sur un circuit fabriqué par IBM [1], contenant 143 tests. Ce circuit est composé de plusieurs modules qui doivent être testés. On parle de catégories de tests : l'amplificateur opérationnel ou bien le LNA, déjà testés, ne sont plus que des modules parmi d'autres qu'il faudra tester.

6.2 Le circuit sous test

Notre cas d'étude est un front-end RF de type zero-IF pour téléphones portables, conçu avec la technologie RFCMOS et fabriqué par IBM. Le schéma du circuit est représenté sur la Figure 6.1.

1. International Business Machines

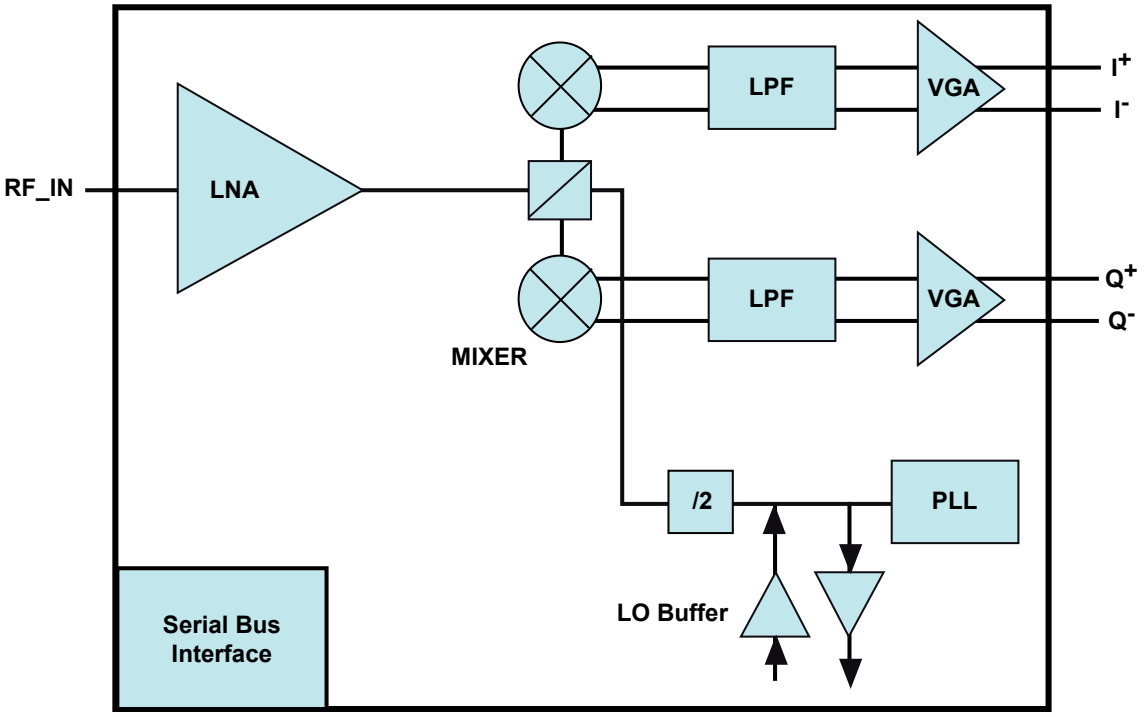


Figure 6.1 – Le Schéma du circuit d'IBM.

Ce circuit est caractérisé par 143 tests répartis en 7 catégories dédiées au test de :

1. la partie numérique ;
2. le courant d'alimentation [2](courant continu DC [3]) ;
3. le convertisseur numérique/analogique (DAC [4]) ;
4. la boucle de verrouillage de phase (PLL [5]) ;
5. le filtre [6] ;
6. le mixer [7] ;
7. l'amplificateur faible bruit (LNA).

Le Tableau 6.1 résume l'ensemble de ces catégories de tests ainsi que le nombre de tests n_i composant chaque catégorie ($i = 0, \ldots, 6$).

2. Supply current
3. Direct Current
4. Digital to Analog Converter
5. Phase Lock Loop
6. Filter
7. Mixer

i	**Catégorie**	**Type**	n_i
0	Partie numérique	Numérique	25
1	Courant d'alimentation	DC	34
2	Convertisseur numérique/analogique (DAC)	Mixte	6
3	Boucle de verrouillage de phase (PLL)	Mixte	7
4	Filtre	RF	20
5	Mixer	RF	43
6	Amplificateur faible bruit (LNA)	RF	8

Tableau 6.1 – Tableau des catégories de tests du circuit d'IBM

L'ensemble des données est composé de 143 tests sur $N = 4450$ circuits, collectées parmi 4 lots de test répartis sur une période de 6 mois. A chaque test correspond des limites de test ou spécifications qui permettent de spécifier l'état de passage ou d'échec d'un circuit par rapport à un test. L'application de ces spécifications a montré que sur le total des 4450 circuits, 4142 circuits sont fonctionnels et 308 circuits sont défectueux. Le circuit est dit défectueux s'il viole au moins un des 143 tests.

Comme notre travail s'inscrit dans le domaine analogique, la catégorie des tests numériques ($n_0 = 25$ tests) ne fera pas partie de notre étude. Une première analyse des données a montré que le test $X73$ du filtre a une valeur constante égale à 20 et aucune dispersion des données comme le montre la Figure 6.2.

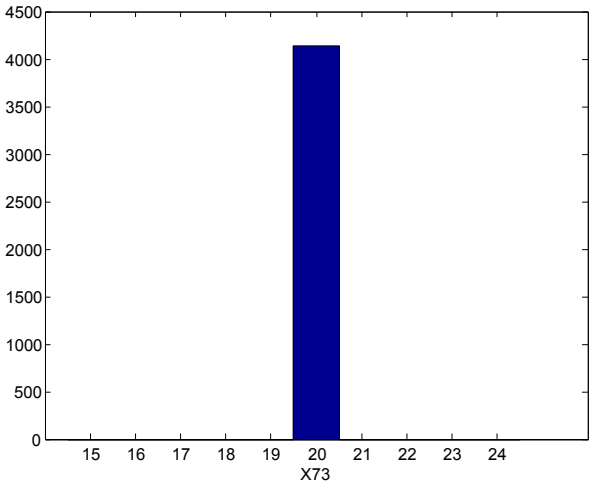


Figure 6.2 – Histogramme du test $X73$ (filtre).

Ce test $X73$ sera écarté de l'ensemble des tests, ce qui réduit le nombre de tests de la catégorie 4 à $n_4 = 19$. En résumé, le circuit à tester est composé de 117 tests répartis en 6 catégories de tests. Pour garder une cohérence avec les données originelles, on va faire référence à chaque test par son numéro (X_j, $j = 26, \ldots, 143$, $j \neq 73$) sans tenir compte des $n_0 = 25$ premiers tests (numériques) ni du test $X73$.

Le test de normalité consiste à appliquer la méthode de la droite de Henry (section 4.2.2.2) à chacun des tests des 6 catégories de tests. Cette méthode validera le choix ou

non d'utliser le modèle multinormal pour chaque catégorie de tests. Il suffit qu'un seul test ne vérifie pas le test de normalité pour invalider le modèle multinormal de toute la catégorie de tests. Comme le nombre de tests est important, on ne représente qu'un seul test de normalité par catégorie de tests sur la Figure 6.3.

L'analyse de la normalité de chacun des tests montre que le nuage des points n'est pas aligné. Cette même observation a été constatée dans le cas d'autres tests dans chacune des 6 catégories de tests. Comme le test de normalité n'est validé pour aucunes des 6 catégories, on va les modéliser par des copules.

6.3 Ordonnancement des tests

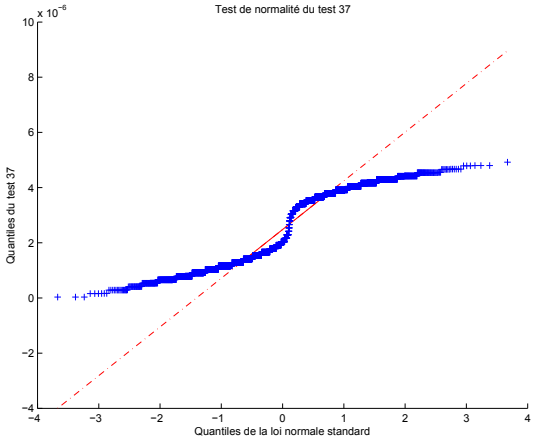
Nous allons ordonner chacune des catégories de tests indépendamment des autres, les tests étant spécifiques à chaque bloc, pour réduire la complexité du problème d'ordonnancement et augmenter la précision des résultats. L'ordonnancement de chacune des 6 catégories de tests se fera suivant le même schéma : modélisation statistique par la méthode des copules, ordonnancement et validation. Une fois le modèle établi et validé, on va générer 30 millions d'instances (critère d'arrêt) pour chaque catégorie de tests. A chaque simulation d'un million d'instances, un ordonnancement des tests est construit sur le cumul des simulations précédentes. Une fois l'ordonnancement des tests établi sur la base des circuits fonctionnels, cet ordre est validé par l'utilisation des données sur les circuits défectueux (écartés lors de la modélisation).

6.3.1 Courant d'alimentation

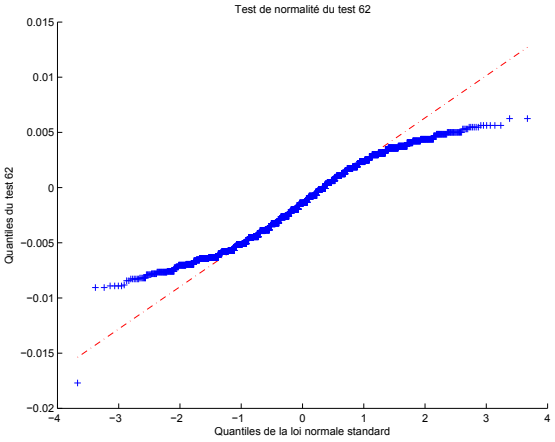
La première catégorie de tests représente l'ensemble des tests de continuité du courant d'alimentation DC, composée de $n_1 = 34$ tests. Pour ordonner les tests de cette catégorie, on va suivre les principales étapes de la simulation d'un échantillon par une copule :

1. ajuster chaque test X de la catégorie par une loi paramétrique usuelle ;
2. utiliser les fonctions de répartition F des lois marginales pour construire les lois uniformes U sur $[0, 1]$;
3. construire la copule correspondant à ces lois uniformes ;
4. générer un nouveau échantillon de la copule ;
5. utiliser les fonctions inverses des fonctions de répartition F^{-1} pour calculer les valeurs des marginales dans l'échantillon généré.

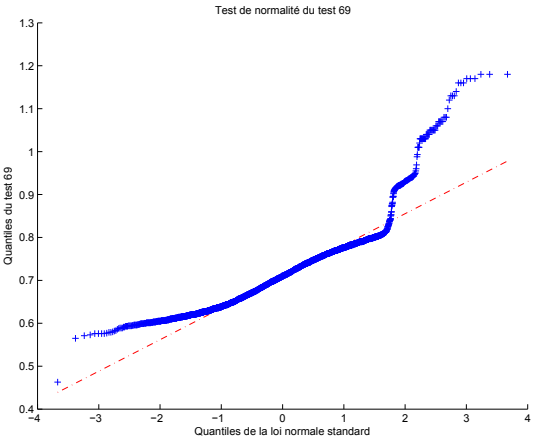
Le Tableau 6.2 représente les spécifications ($[a_1, a_2]$), issues du cahier des charges des 34 tests, ainsi que les distributions marginales ajustant chaque test.



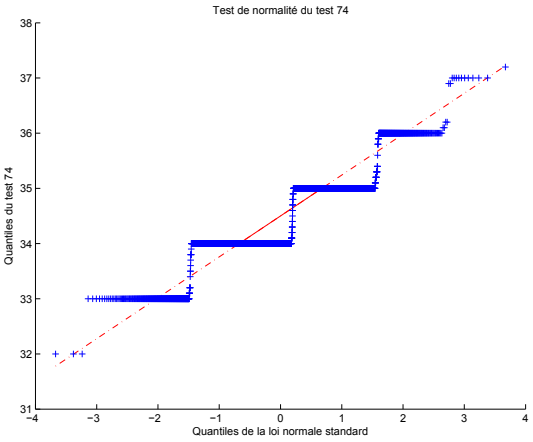
(a) X37 - catégorie 1



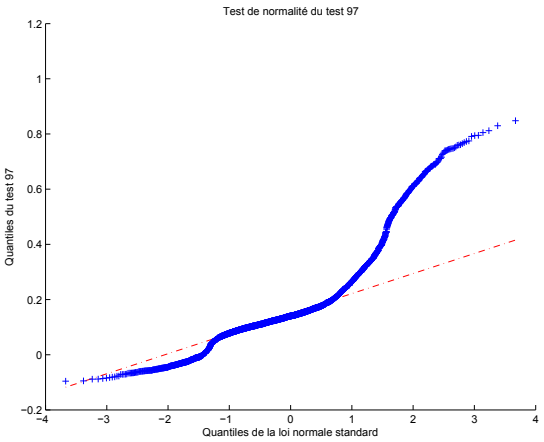
(b) X62 - catégorie 2



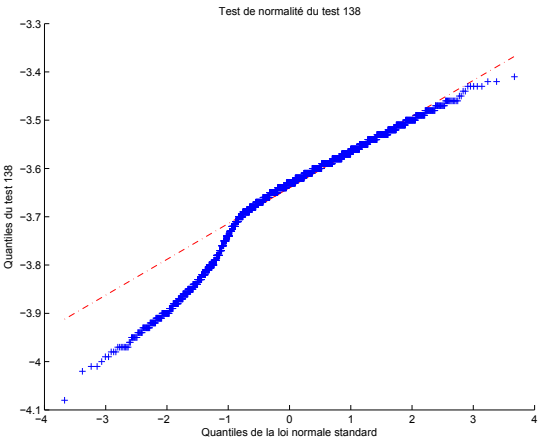
(c) X69 - catégorie 3



(d) X74 - catégorie 4



(e) X97 - catégorie 5



(f) X138 - catégorie 6

Figure 6.3 – Test de normalité d'un test par catégorie.

N°	Test	Nom	Spécifications		Distribution marginale
			a_1	a_2	
1	X26	LEAKAGE 1	-2e-6	2e-6	Non paramétrique
2	X27	LEAKAGE 2	-2e-6	2e-6	Normale
3	X28	LEAKAGE 3	-2e-6	2e-6	Valeur extrême généralisée
4	X29	Idd1	-1e-5	5e-5	Normale
5	X30	Idd2	-1e-5	5e-5	Normale
6	X31	Idd3	-1e-5	5e-5	Valeur extrême généralisée
7	X32	Idd4	-1e-5	5e-5	Valeur extrême généralisée
8	X33	Idd5	-1e-5	5e-5	Valeur extrême généralisée
9	X34	Idd6	-1e-5	5e-5	Valeurs extrêmes
10	X35	Idd7	-1e-5	5e-5	Normale
11	X36	Idd8	-1e-5	5e-5	Valeurs extrêmes
12	X37	Idd9	-1e-5	5e-5	Normale
13	X38	Idd10	-1e-5	5e-5	Normale
14	X39	Idd11	-1e-5	6e-5	Valeur extrême généralisée
15	X40	Idd12	-1e-5	1e-4	Normale
16	X41	Idd13	-1e-5	5e-5	Valeur extrême généralisée
17	X42	Idd14	0	5e-3	Valeur extrême généralisée
18	X43	Idd15	6e-3	0.0135	Valeur extrême généralisée
19	X44	Idd16	0	0.0155	Non paramétrique
20	X45	Idd17	0	0.01	Non paramétrique
21	X46	Idd18	0	0.01	Non paramétrique
22	X47	Idd19	0	0.025	Normale
23	X48	Idd20	0	6e-3	Normale
24	X49	Idd21	0	0.014	Normale
25	X50	Idd22	0.048	0.064	Non paramétrique
26	X51	Idd23	0.01	0.055	Valeur extrême généralisée
27	X52	Idd24	0	0.02	Valeurs extrêmes
28	X53	Idd25	0	0.016	Valeurs extrêmes
29	X54	Idd26	0	5e-3	Valeur extrême généralisée
30	X55	Idd27	0	5e-3	Valeur extrême généralisée
31	X56	Idd28	-1e-5	1e-3	Normale
32	X57	Idd29	-3e-5	1e-3	Normale
33	X58	Idd30	-3e-5	1e-3	Valeur extrême généralisée
34	X59	Idd31	-3e-5	1e-3	Normale

Tableau 6.2 – Spécifications et ajustement des tests de courant.

La première ligne du Tableau 6.2 représente le test $X26$, dont les spécifications sont $a_1 = -2e-6$ et $a_2 = 2e-6$. Ce premier test est ajusté par une distribution marginale non paramétrique. Par contre le deuxième test $X27$ est ajusté par une distribution normale et ses spécifications sont $[-2e-6\ ,\ 2e-6]$ et ainsi de suite pour les autres tests.

Une analyse des échantillons simulés suivant la copule gaussienne, dont les distributions marginales sont données dans le Tableau 6.2, montre que seulement les tests numéro 16 ($X41$) et 25 ($X50$) détectent des circuits défectueux, alors que les 32 autres tests ne détectent aucun circuit défectueux. Cette remarque nous a permis de réduire l'ensemble

des tests à ordonner à seulement 2 tests au lieu des 34 d'origine. Ceci est très important car il permet d'utiliser la méthode optimale du branch and bound au lieu des algorithmes génétiques ou des méthodes de recherche flottante. Les résultats de l'application de la méthode d'ordonnancement des tests de la Figure 5.4 sur les tests de la première catégorie sont résumés dans le Tableau 6.3

Nombre de	**Méthode statistique**			**Heuristique de capabilité**			**Heuristique corrélation**	
tests	**tests**	$FC_{exp}(\%)$	D **à** 95%	**tests**	C_{pk}	$FC_{exp}(\%)$	**tests**	$FC_{exp}(\%)$
0	Aucun	0	[1642.22 ;1650.04]	0	0	0	0	0
1	16	4.74	[141.27 ; 143.55]	16	0.67	4.74	9	15.64
2	25	81.52	[0 ; 0]	25	1.62	81.52	3	18.01
3	34	81.99	[0 ; 0]	18	2.69	85.78	27	18.48
4	33	82.46	[0 ; 0]	15	2.89	91	15	53.55
5	32	82.46	[0 ; 0]	12	3.28	91	7	54.98
6	31	82.46	[0 ; 0]	24	3.49	91	34	54.98
7	30	82.46	[0 ; 0]	28	5.16	91	23	57.82
8	29	82.46	[0 ; 0]	22	5.31	91	32	57.82
9	28	82.46	[0 ; 0]	27	5.74	91	13	59.24
10	27	82.46	[0 ; 0]	19	5.80	91	16	59.24
11	26	82.46	[0 ; 0]	30	6.34	91	20	59.24
12	24	82.46	[0 ; 0]	26	6.36	91	4	65.40
13	23	82.46	[0 ; 0]	34	6.39	91	24	65.40
14	22	82.46	[0 ; 0]	31	6.45	91	11	65.40
15	21	82.46	[0 ; 0]	8	6.97	91.94	8	68.25
16	20	82.46	[0 ; 0]	5	7.01	91.94	6	69.67
17	19	82.46	[0 ; 0]	14	7.29	100	33	69.67
18	18	86.26	[0 ; 0]	7	7.35	100	21	69.67
19	17	86.26	[0 ; 0]	9	7.66	100	10	70.62
20	15	91	[0 ; 0]	4	7.72	100	19	70.62
21	14	100	[0 ; 0]	6	8.68	100	1	70.62
22	13	100	[0 ; 0]	17	8.87	100	14	71.09
23	12	100	[0 ; 0]	10	9.43	100	18	78.67
24	11	100	[0 ; 0]	2	9.85	100	28	78.67
25	10	100	[0 ; 0]	1	9.99	100	2	78.67
26	9	100	[0 ; 0]	32	10.60	100	5	78.67
27	8	100	[0 ; 0]	13	11.33	100	30	78.67
28	7	100	[0 ; 0]	20	11.91	100	12	79.15
29	6	100	[0 ; 0]	33	13.08	100	22	79.15
30	5	100	[0 ; 0]	21	15.85	100	31	79.15
31	4	100	[0 ; 0]	29	16.33	100	29	79.15
32	3	100	[0 ; 0]	3	16.86	100	26	79.15
33	2	100	[0 ; 0]	23	21.07	100	17	79.15
34	1	100	[0 ; 0]	11	500.19	100	25	100

Tableau 6.3 – Ordonnancement des tests du courant d'alimentation.

Le Tableau 6.3 présente une comparaison entre les résultats de la méthode d'ordonnancement des tests de la catégorie 1 et l'ordre obtenu pas les heuristiques simples. Pour chacune des méthodes, on a présenté l'ordonnancement des tests avec la couverture de fautes expérimentale $FC_{exp}(\%)$. L'intervalle de confiance du taux de défauts fixé à 95% (D) est calculé pour l'ordonnancement effectué par la méthode d'ordonnancement des tests.

Une analyse des résultats montre que l'ensemble comportant les deux tests 16 et 25 garantit un taux de défauts paramétriques nul. Quand à la couverture des circuits défectueux, ces deux tests assurent une couverture de 81.52% des circuits défectueux. Le même ensemble de tests est détecté par l'heuristique de capabilité. Par contre, l'heuristique basée sur la corrélation n'assure qu'une faible couverture de 18.01% des circuits défectueux avec les tests numéros 9 et 3.

Utilisons maintenant les circuits défectueux, non utilisés précédemment, pour obtenir un ensemble minimal de tests couvrant 100% des circuits défectueux. Ceci est réalisé par l'application de la méthode de sélection (Figure 5.5) sur les tests de la catégorie 1. L'ensemble minimal couvrant 100% des circuits défectueux est l'ensemble des tests $\{14, 15, 25\}$. On remarque que notre méthode, n'utilisant aucun circuit défectueux, a pu trouver un test (25) parmi les 3 tests composant cet ensemble minimal. En plus, le test numéro 25 est le plus important car il assure à lui seul une couverture des circuits défectueux de 80.57%.

Maintenant qu'on dispose d'un ensemble de tests $\{16, 25\}$ détectant toutes les fautes paramétriques et d'un ensemble minimal $\{14, 15, 25\}$ assurant une couverture des circuits défectueux de 100%, nous allons appliquer la méthode de sélection et d'ordonnancement (Figure 5.6) sur les deux ensembles précédents pour former un nouvel ensemble garantissant la détection de toutes les fautes paramétriques et une couverture des circuits défectueux de 100%. Les résultats de cette expérimentation sont illustrés dans le Tableau 6.4.

Nombre de tests	**Séquence de tests**	**FC (%) par test**	**FC (%) cumulé**	**Intervalle de confiance du taux de défauts paramétriques à 95%**
1	25	80.57	80.57	[1499.81 ; 1508.48]
2	14	69.19	99.05	[1499.81 ; 1508.48]
3	15	49.76	100	[1499.81 ; 1508.48]
4	16	4.74	100	[0 ; 0]

Tableau 6.4 – Ensemble compact et ordonné des tests du courant d'alimentation.

Une analyse des résultats du Tableau 6.4 montre que le premier test à appliquer est le test numéro 25 qui assure une couverture des circuits défectueux de 80.57%. Ensuite, vient le test numéro 14 qui augmente la couverture à 99.05%. La couverture des circuits défectueux à 100% est assurée par les 3 tests $\{25, 14, 15\}$ alors que l'ajout du dernier test numéro 16 permet la détection de toutes les fautes paramétriques avec un intervalle de confiance du taux de défauts nul. En résumé, l'ensemble des tests $\{25, 14, 15, 16\}$ garantit à

la fois une couverture des circuits défectueux à 100% et un taux de défauts paramétriques nul.

6.3.2 Convertisseur numérique/analogique (DAC)

L'ensemble des tests du convertisseur numérique/analogique est composé de $n_2 = 6$ tests. Le Tableau 6.5 représente les spécifications des 6 tests ainsi que leurs distributions marginales d'ajustement.

N°	Test	Nom	Spécifications		Distribution marginale
			a_1	a_2	
1	X60	I1_DAC	1	247	Normale
2	X61	I2_DAC	-0.019	190	Valeur extrême généralisée
3	X62	I3_DAC	-0.07	0.07	Valeur extrême généralisée
4	X63	I4_DAC	1	247	Valeurs extrêmes
5	X64	I5_DAC	-0.019	0.019	Valeur extrême généralisée
6	X65	I6_DAC	-0.07	0.07	Valeur extrême généralisée

Tableau 6.5 – Spécifications et ajustement des tests du DAC.

Une fois un échantillon de 30 millions d'instances simulé, l'ordonnancement des tests obtenu est illustré dans le Tableau 6.6.

Nombre de tests	Méthode statistique			Heuristique de capabilité			Heuristique corrélation	
	tests	$FC_{exp}(\%)$	D à 95%	tests	C_{pk}	$FC_{exp}(\%)$	tests	$FC_{exp}(\%)$
0	Aucun	0	[0 ; 0]	0	0	0	0	0
1	1	53.97	[0 ; 0]	1	3.46	53.97	4	63.49
2	2	53.97	[0 ; 0]	4	3.81	90.48	3	71.43
3	3	60.32	[0 ; 0]	5	5.01	90.48	5	71.43
4	4	96.83	[0 ; 0]	2	5.15	90.48	1	96.83
5	5	96.83	[0 ; 0]	6	6.74	96.83	6	100
6	6	100	[0 ; 0]	3	6.89	100	2	100

Tableau 6.6 – Ordonnancement des tests du convertisseur numérique/analogique (DAC).

Ce résultat montre que l'ordre d'élimination des tests n'est pas important car aucun circuit défectueux n'est généré parmi l'échantillon des 30 millions d'instances. Ce résultat est dû à la largeur des intervalles des spécifications qui sont très largement supérieurs à l'étendue définie par la variation des données. Dans ce cas, on ne peut pas ordonner les tests par la méthode d'ordonnancement des tests. On prendra donc comme ordre des tests, l'ordre donnée par l'heuristique basée sur la corrélation car le premier test, 4, détecte 63.49% des circuits défectueux alors que l'heuristique de capabilité ne détecte que 53.97% des circuits défectueux. En plus, l'heuristique basée sur la corrélation atteint 100% de la couverture au $5^{ième}$ test alors que l'heuristique de capabilité ne l'atteint qu'au dernier test.

L'utilisation de la méthode de sélection donne comme résultat l'ensemble des tests $\{1, 4, 3, 6\}$. Comme l'ensemble des tests de l'ordre paramétrique est vide, l'ensemble $\{1, 4, 3, 6\}$ minimal couvrant 100% des circuits défectueux sera l'unique entrée de la méthode de sélection et d'ordonnancement. Les résultats sont illustrés dans le Tableau 6.7.

Nombre de tests	**Séquence de tests**	**FC (%) par test**	**FC (%) cumulé**	**Intervalle de confiance du taux de défauts paramétriques à** 95%
1	4	63.49	63.49	[0 ; 0]
2	1	53.97	90.48	[0 ; 0]
3	6	28.57	96.83	[0 ; 0]
4	3	20.63	100	[0 ; 0]

Tableau 6.7 – Ensemble compact et ordonné des tests du convertisseur numérique/analogique (DAC).

Le test le plus important en termes de couverture des circuits défectueux est le test numéro 4 avec une couverture de 63.49%. Il sera suivi successivement par les tests 1, 6 et 3 pour atteindre une couverture des circuits défectueux de 100%. Quant aux fautes paramétriques, elles ne sont pas générées dans cette catégorie car les limites de spécification sont plus larges que l'étendue de la variation des tests composant cette catégorie. L'ensemble des tests $\{4, 1, 6, 3\}$ garantit donc à la fois une couverture des circuits défectueux à 100% et un taux de défauts paramétriques nul.

6.3.3 Boucle de verrouillage de phase (PLL)

L'ensemble des tests de la boucle de verrouillage de phase (PLL) est composé de $n_3 = 7$ tests. Le Tableau 6.8 représente leurs spécifications ainsi que leurs distributions marginales d'ajustement.

N°	**Test**	**Nom**	**Spécifications**		**Distribution marginale**
			$\mathbf{a_1}$	$\mathbf{a_2}$	
1	X66	tune_code_1	10	48	Normale
2	X67	tune_code_2	10	48	Normale
3	X68	tune_code_3	10	48	Normale
4	X69	voltage 1	0.1	1.8	Normale
5	X70	voltage 2	0.1	1.8	Valeurs extrêmes
6	X71	voltage 3	0.1	1.8	Normale
7	X72	LO_amplitude	-19	99	Non paramétrique

Tableau 6.8 – Spécifications et ajustement des tests de la PLL.

Une fois un échantillon de 30 millions d'instances simulé, l'ordonnancement des tests obtenu est illustré dans le Tableau 6.9.

Nombre de tests	Méthode statistique			Heuristique de capabilité			Heuristique corrélation	
	tests	$FC_{exp}(\%)$	D à 95%	tests	C_{pk}	$FC_{exp}(\%)$	tests	$FC_{exp}(\%)$
0	Aucun	0	[1.16 ; 1.35]	0	0	0	0	0
1	1	98.77	[0 ; 0]	3	2.32	97.55	7	98.77
2	7	100	[0 ; 0]	4	2.61	98.77	6	100
3	6	100	[0 ; 0]	5	2.71	98.77	4	100
4	5	100	[0 ; 0]	2	2.99	99.39	5	100
5	4	100	[0 ; 0]	6	3.14	99.39	1	100
6	3	100	[0 ; 0]	1	3.70	99.39	3	100
7	2	100	[0 ; 0]	7	8.51	100	2	100

Tableau 6.9 – Ordonnancement des tests de la boucle de verrouillage de phase (PLL).

L'analyse des résultats du Tableau 6.9 montre que le test numéro 1 permet à lui seul de détecter toutes les fautes paramétriques avec une couverture des circuits défectueux de 98.77%. Cette même couverture est obtenue par l'heuristique basée sur la corrélation, ce qui montre que les tests numéros 1 et 7 ont le même comportement. Par contre l'heuristique de capabilité propose le test numéro 3 avec une couverture des circuits défectueux de 97.55%.

En utilisant les circuits défectueux, la méthode de sélection donne $\{7, 1\}$ comme un ensemble minimal garantissant une couverture des circuits défectueux de 100%. Le test 1 fait partie aussi des tests repérés par la méthode d'ordonnancement des tests qui n'utilise aucun circuit défectueux dans sa construction. La méthode d'ordonnancement des tests a donc trouvé la moitié des tests couvrant 100% des circuits défectueux, alors qu'elle n'utilise pas de données sur les défectueux dans son fonctionnement.

La combinaison de l'ordre paramétrique, composé du seul test 1, et de l'ensemble minimal couvrant 100% des circuits défectueux $\{7, 1\}$ est illustré dans le Tableau 6.10.

Nombre de tests	Séquence de tests	FC (%) par test	FC (%) cumulé	Intervalle de confiance du taux de défauts paramétriques à 95%
1	1	98.77	98.77	[0 ; 0]
2	7	98.77	100	[0 ; 0]

Tableau 6.10 – Ensemble compact et ordonné de la boucle de verrouillage de phase (PLL).

Le Tableau 6.10 montre clairement qu'il suffit de tester les deux tests 1 et 7 pour assurer à la fois une couverture des circuits défectueux de 100% et un taux de défauts paramétriques nul.

6.3.4 Filtre

L'ensemble des tests du filtre est composé de $n_4 = 19$ tests. Le Tableau 6.11 représente les spécifications des différents tests de cette catégorie ainsi que leurs distributions marginales d'ajustement.

N°	Test	Nom	Spécifications		Distribution marginale
			a_1	a_2	
1	X74	Filter_test1	10	60	Normale
2	X75	Filter_test2	0	3.9	Valeurs extrêmes
3	X76	Filter_test3	0	3.9	Normale
4	X77	Filter_test4	3	25	Valeur extrême généralisée
5	X78	Filter_test5	3	25	Valeur extrême généralisée
6	X79	Filter_test6	0	0.65	Valeurs extrêmes
7	X80	Filter_test7	0	0.65	Normale
8	X81	Filter_test8	-999	-1	Valeur extrême généralisée
9	X82	Filter_test9	-999	-1	Valeur extrême généralisée
10	X83	Filter_test10	-999	-15.7	Normale
11	X84	Filter_test11	-999	-1	Non paramétrique
12	X85	Filter_test12	-999	-1	Valeur extrême généralisée
13	X86	Filter_test13	-2	999	Valeur extrême généralisée
14	X87	Filter_test14	-2	999	Valeur extrême généralisée
15	X88	Filter_test15	-0.49	0	Valeur extrême généralisée
16	X89	Filter_test16	-2	999	Valeur extrême généralisée
17	X90	Filter_test17	-2	999	Valeur extrême généralisée
18	X91	Filter_test18	12	999	Valeur extrême généralisée
19	X92	Filter_test19	13	99	Valeur extrême généralisée

Tableau 6.11 – Spécifications et ajustement des tests du filtre.

Une fois qu'un échantillon de 30 millions d'instances est simulé, l'ordonnancement des tests obtenu est illustré dans le Tableau 6.12.

Nombre de tests	Méthode statistique			Heuristique de capabilité			Heuristique corrélation	
	tests	$FC_{exp}(\%)$	D à 95%	tests	C_{pk}	$FC_{exp}(\%)$	tests	$FC_{exp}(\%)$
0	Aucun	0	[17776.11 ;17793.55]	0	0	0	0	0
1	5	51.89	[1785.64 ;1796.04]	7	0.76	0.54	13	0
2	4	69.19	[1511.47 ;1520.14]	6	0.77	1.08	1	48.65
3	10	99.46	[0.70 ; 0.90]	5	1.08	52.43	7	49.19
4	7	99.46	[0 ; 0]	4	1.10	69.73	10	94.05
5	6	99.46	[0 ; 0]	10	1.78	99.46	3	96.76
6	19	99.46	[0 ; 0]	15	3.14	99.46	14	96.76
7	18	99.46	[0 ; 0]	19	3.41	99.46	19	96.76
8	17	99.46	[0 ; 0]	12	4.72	99.46	2	96.76
9	16	99.46	[0 ; 0]	8	4.72	99.46	9	96.76
10	15	99.46	[0 ; 0]	11	5.33	99.46	6	96.76
11	14	99.46	[0 ; 0]	18	5.46	99.46	8	96.76
12	13	99.46	[0 ; 0]	2	5.51	99.46	15	96.76
13	12	99.46	[0 ; 0]	9	5.77	99.46	11	96.76
14	11	99.46	[0 ; 0]	3	6.81	99.46	18	96.76
15	9	99.46	[0 ; 0]	17	10.10	99.46	12	96.76
16	8	99.46	[0 ; 0]	13	10.88	99.46	4	99.46

Suite page suivante

Nombre de tests	Méthode statistique			Heuristique de capabilité			Heuristique corrélation	
	tests	$FC_{exp}(\%)$	D à 95%	tests	C_{pk}	$FC_{exp}(\%)$	tests	$FC_{exp}(\%)$
17	3	99.46	[0 ; 0]	1	11.39	100	5	100
18	2	99.46	[0 ; 0]	16	20.16	100	17	100
19	1	100	[0 ; 0]	14	23.17	100	16	100

Tableau 6.12 – Ordonnancement des tests du filtre.

Le Tableau 6.12 montre que les 4 premiers tests $\{5, 4, 10, 7\}$ assurent la détection de toutes les fautes paramétriques ainsi qu'une couverture des circuits défectueux de 99.46%. Alors que les deux heuristiques ne parviennent qu'à atteindre 94.05% pour l'heuristique basée sur la corrélation et 69.73% pour l'heuristique de la capabilité.

L'application de la méthode de sélection sur les tests de cette catégorie aboutit à l'ensemble des tests $\{10, 5, 1\}$ couvrant à 100% les circuits défectueux. Encore une fois la méthode d'ordonnancement des tests a obtenu 2 tests sur les 3 formant l'ensemble minimal de tests avec une couverture de 100%.

La combinaison de l'ordre paramétrique composé des tests $\{5, 4, 10, 7\}$ et l'ensemble minimal couvrant 100% des circuits défectueux $\{10, 5, 1\}$ est illustré dans le Tableau 6.13.

Nombre de tests	Séquence de tests	FC (%) par test	FC (%) cumulé	Intervalle de confiance du taux de défauts paramétriques à 95%
1	10	93.51	93.51	[16304.71 ; 16326.19]
2	5	51.89	99.46	[275.18 ; 277.71]
3	1	48.65	100	[275.18 ; 277.71]
4	4	47.57	100	[0.70 ; 0.90]
5	7	0.54	100	[0 ; 0]

Tableau 6.13 – Ensemble compact et ordonné du filtre.

A noter qu'il faut ajouter le test 1 pour la détection des circuits défectueux, mais que ce test à un C_{pk} très grand (23.17).

Le Tableau 6.13 montre que l'ensemble des tests $\{10, 5, 1, 4, 7\}$ garantit à la fois une couverture des circuits défectueux de 100% et un taux de défauts paramétriques nul.

6.3.5 Mélangeur (Mixer)

Le mixer est le plus grand bloc du circuit sous test en termes de nombre de tests. Il est composé de $n_5 = 43$ tests représentés dans le Tableau 6.11 avec leurs spécifications et leurs distributions marginales d'ajustement.

N°	Test	Nom	Spécifications		Distribution marginale
			a_1	a_2	
1	X93	Mixer_test1	29.3	33.7	Valeur extrême généralisée
2	X94	Mixer_test2	29.3	33.7	Valeur extrême généralisée
3	X95	Mixer_test3	29.3	33.7	Non paramétrique
4	X96	Mixer_test4	24.8	999	Normale
5	X97	Mixer_test5	-999	1.9	Non paramétrique
6	X98	Mixer_test6	-999	7	Normale
7	X99	Mixer_test7	45.3	49.7	Non paramétrique
8	X100	Mixer_test8	45.3	49.7	Non paramétrique
9	X101	Mixer_test9	45.3	49.7	Non paramétrique
10	X102	Mixer_test10	35	999	Non paramétrique
11	X103	Mixer_test11	35	999	Valeur extrême généralisée
12	X104	Mixer_test12	-1	64	Normale
13	X105	Mixer_test13	-60	60	Normale
14	X106	Mixer_test14	35	999	Normale
15	X107	Mixer_test15	35	999	Normale
16	X108	Mixer_test16	-1	64	Normale
17	X109	Mixer_test17	-0.6	60	Non paramétrique
18	X110	Mixer_test18	55	999	Non paramétrique
19	X111	Mixer_test19	55	999	Valeurs extrêmes
20	X112	Mixer_test20	35	999	Non paramétrique
21	X113	Mixer_test21	35	999	Non paramétrique
22	X114	Mixer_test22	4.2	999	Normale
23	X115	Mixer_test23	4.2	999	Normale
24	X116	Mixer_test24	4.2	999	Normale
25	X117	Mixer_test25	20	999	Valeur extrême généralisée
26	X118	Mixer_test26	20	999	Normale
27	X119	Mixer_test27	0.2	999	Valeur extrême généralisée
28	X120	Mixer_test28	-1.5	1.5	Non paramétrique
29	X121	Mixer_test29	-1.5	1.5	Normale
30	X122	Mixer_test30	-1.5	1.5	Non paramétrique
31	X123	Mixer_test31	-1.5	1.5	Normale
32	X124	Mixer_test32	-1.5	1.5	Non paramétrique
33	X125	Mixer_test33	-1.5	1.5	Non paramétrique
34	X126	Mixer_test34	-1.5	1.5	Non paramétrique
35	X127	Mixer_test35	-1.5	1.5	Non paramétrique
36	X128	Mixer_test36	-999	12.9	Valeur extrême généralisée
37	X129	Mixer_test37	-999	12.9	Normale
38	X130	Mixer_test38	-999	12.9	Normale
39	X131	Mixer_test39	-999	31.8	Normale
40	X132	Mixer_test40	-999	31.8	Normale
41	X133	Mixer_test41	-999	31.8	Normale
42	X134	Mixer_test42	-999	-61	Non paramétrique
43	X135	Mixer_test43	-999	-30	Non paramétrique

Tableau 6.14 – Spécifications et ajustement des tests du mixer.

L'ordonnancement des tests est illustré dans le Tableau 6.15.

Nombre de tests	**Méthode statistique**			**Heuristique de capabilité**			**Heuristique corrélation**	
	tests	$FC_{exp}(\%)$	D **à** 95%	**tests**	C_{pk}	$FC_{exp}(\%)$	**tests**	$FC_{exp}(\%)$
0	Aucun	0	[34259.00 ;34282.92]	0	0	0	0	0
1	12	0	[18821.08 ;18851.11]	19	0.62	79.34	43	0
2	8	84.71	[10886.31 ;10899.41]	18	0.65	79.75	14	79.75
3	19	84.71	[5908.86 ;5924.66]	25	0.69	80.17	25	81.40
4	18	84.71	[2858.67 ;2873.53]	20	0.71	81.40	20	83.06
5	3	85.54	[1428.15 ;1438.43]	21	0.80	83.06	35	84.30
6	20	86.78	[952.47 ; 957.68]	27	0.86	83.47	40	84.71
7	31	87.19	[507.74 ; 512.97]	22	0.96	85.12	28	84.71
8	4	95.04	[227.38 ; 229.63]	26	0.97	85.54	26	85.12
9	25	95.04	[85.23 ; 86.35]	24	1.05	85.54	42	85.12
10	21	95.04	[49.29 ; 50.23]	23	1.07	86.36	10	85.12
11	29	95.45	[21.68 ; 24.20]	31	1.11	86.36	1	87.19
12	2	95.45	[20.38 ; 21.05]	4	1.14	94.63	21	87.19
13	30	95.45	[6.06 ; 6.37]	30	1.24	94.63	31	87.19
14	34	95.87	[1.03 ; 1.19]	8	1.31	96.28	16	87.19
15	22	96.69	[0.09 ; 0.13]	29	1.35	96.28	5	87.19
16	9	96.69	[0.09 ; 0.13]	2	1.38	96.28	6	87.19
17	1	96.69	[0.09 ; 0.13]	11	1.41	96.28	32	88.02
18	24	96.69	[0.09 ; 0.13]	14	1.41	96.69	39	88.02
19	23	97.11	[0 ; 0]	15	1.41	96.69	11	88.02
20	43	97.11	[0 ; 0]	10	1.47	96.69	29	88.43
21	42	97.11	[0 ; 0]	1	1.63	97.11	15	88.43
22	41	97.11	[0 ; 0]	9	1.64	97.11	22	89.67
23	40	97.11	[0 ; 0]	3	1.71	97.11	2	90.08
24	39	97.11	[0 ; 0]	7	1.87	97.11	34	90.08
25	38	97.52	[0 ; 0]	28	1.92	97.11	12	90.08
26	37	97.93	[0 ; 0]	35	2.04	97.93	36	90.50
27	36	97.93	[0 ; 0]	33	2.14	98.76	30	90.50
28	35	98.76	[0 ; 0]	6	2.23	98.76	33	90.91
29	33	99.59	[0 ; 0]	34	2.25	99.17	27	90.91
30	32	99.59	[0 ; 0]	32	2.48	99.17	37	91.32
31	28	99.59	[0 ; 0]	39	2.57	99.17	18	91.32
32	27	99.59	[0 ; 0]	37	2.70	100	4	99.17
33	26	99.59	[0 ; 0]	36	2.88	100	23	99.59
34	17	99.59	[0 ; 0]	40	2.92	100	8	100
35	16	99.59	[0 ; 0]	16	3.07	100	19	100
36	15	100	[0 ; 0]	38	3.28	100	3	100
37	14	100	[0 ; 0]	41	3.61	100	41	100
38	13	100	[0 ; 0]	12	3.79	100	13	100
39	11	100	[0 ; 0]	43	4.08	100	7	100
40	10	100	[0 ; 0]	5	4.09	100	24	100
41	7	100	[0 ; 0]	17	5.16	100	38	100

Suite page suivante

Nombre de tests	Méthode statistique			Heuristique de capabilité			Heuristique corrélation	
	tests	$FC_{exp}(\%)$	D à 95%	tests	C_{pk}	$FC_{exp}(\%)$	tests	$FC_{exp}(\%)$
42	6	100	[0 ; 0]	13	6.14	100	9	100
43	5	100	[0 ; 0]	42	6.38	100	17	100

Tableau 6.15 – Ordonnancement des tests du mixer.

Les résultats montrent qu'il suffit de tester 19 tests pour garantir un taux de défaut paramétrique nul, ensuite pour le reste des tests l'ordre n'est pas important car le taux de défauts paramétriques est nul. Donc parmi les 43 tests de cette catégorie, uniquement 19 tests sont importants et leur ordre d'élimination est bien établi par la méthode d'ordonnancement. Le test de ces 19 tests permet aussi d'atteindre une couverture des circuits défectueux de 97.11%, une valeur plus grande que celle obtenue par les deux heuristiques simples.

L'application de la méthode de sélection a permis de retrouver l'ensemble minimal couvrant 100% des circuits défectueux qui est $\{35, 4, 23, 8, 20, 37, 33, 22, 32, 1, 2\}$. Ainsi, la méthode d'ordonnancement a pu retrouver 6 tests parmi les 11 tests formant l'ensemble minimal couvrant 100% des circuits défectueux.

La combinaison de l'ordre paramétrique et de l'ensemble minimal couvrant 100% des circuits défectueux est illustré dans le Tableau 6.16.

Nombre de tests	Séquence de tests	FC (%) par test	FC (%) cumulé	Intervalle de confiance du taux de défauts paramétriques à 95%
1	4	88.43	88.43	[33975.74 ; 34001.96]
2	8	84.71	92.56	[26158.40 ; 26167.02]
3	9	84.30	92.56	[26158.40 ; 26167.02]
4	2	83.47	92.98	[26139.91 ; 26148.69]
5	3	83.47	93.39	[24761.38 ; 24770.77]
6	23	83.06	95.45	[24761.27 ; 24770.66]
7	22	82.23	96.28	[24760.29 ; 24769.75]
8	24	82.23	96.28	[24760.29 ; 24769.75]
9	37	81.82	97.11	[24760.29 ; 24769.75]
10	30	81.40	97.11	[24744.81 ; 24753.78]
11	1	80.99	97.11	[24744.81 ; 24753.78]
12	21	80.99	97.11	[24709.02 ; 24717.88]
13	20	80.17	97.93	[24241.55 ; 24253.24]
14	18	79.34	97.93	[21168.38 ; 21179.39]
15	19	79.34	97.93	[16193.67 ; 16203.26]
16	25	78.93	97.93	[16053.16 ; 16061.83]
17	29	74.38	97.93	[16025.52 ; 16034.43]
18	35	71.90	98.76	[16025.52 ; 16034.43]
19	31	68.60	98.76	[15590.18 ; 15598.96]
20	32	68.60	99.59	[15590.18 ; 15598.96]
21	34	68.18	99.59	[15585.07 ; 15594.19]

Suite page suivante

Nombre de tests	Séquence de tests	FC (%) par test	FC (%) cumulé	Intervalle de confiance du taux de défauts paramétriques à 95%
22	33	67.36	100	[15585.07 ; 15594.19]
23	12	0	100	[0 ; 0]

Tableau 6.16 – Ensemble compact et ordonné du mixer.

6.3.6 Amplificateurs faible bruit (LNA)

L'ensemble des tests de l'amplificateur faible bruit LNA est composé de $n_6 = 8$ tests. Le Tableau 6.8 représente les spécifications de ces 8 tests ainsi que leurs distributions marginales d'ajustement.

N°	Test	Nom	Spécifications		Distribution marginale
			a_1	a_2	
1	X136	LNA_test1	15.4	17.6	Valeurs extrêmes
2	X137	LNA_test2	3.3	5.3	Non paramétrique
3	X138	LNA_test3	-5.6	-3.4	Non paramétrique
4	X139	LNA_test4	-19.6	-16.4	Normal
5	X140	LNA_test5	0	2	Normal
6	X141	LNA_test6	0	5.8	Normal
7	X142	LNA_test7	7.8	99	Valeur extrême généralisée
8	X143	LNA_test8	6.4	99	Normal

Tableau 6.17 – Spécifications et ajustement des tests du LNA.

Après une trentaine de simulations d'un million d'instances, l'ordonnancement des tests est illustré dans le Tableau 6.18.

Nombre de tests	Méthode statistique			Heuristique de capabilité			Heuristique corrélation	
	tests	$FC_{exp}(\%)$	D à 95%	tests	C_{pk}	$FC_{exp}(\%)$	tests	$FC_{exp}(\%)$
0	Aucun	0	[151.57 ; 152.75]	0	0	0	0	0
1	3	76.98	[24.73 ; 26.03]	3	0.85	76.98	8	9.35
2	1	82.73	[0.01 ; 0.02]	1	1.61	82.73	2	77.70
3	5	99.28	[0 ; 0]	5	1.79	99.28	4	79.14
4	8	100	[0 ; 0]	2	1.95	99.28	6	79.86
5	7	100	[0 ; 0]	4	2.21	100	5	95.68
6	6	100	[0 ; 0]	6	2.51	100	1	97.12
7	4	100	[0 ; 0]	7	4.17	100	3	100
8	2	100	[0 ; 0]	8	5.76	100	7	100

Tableau 6.18 – Ordonnancement des tests du LNA.

Les résultats montrent que l'ensemble des 3 tests $\{3, 1, 5\}$ détecte toutes les fautes paramétriques et assure une couverture des circuits défectueux de 99.28%. Cette même couverture est obtenue par l'heuristique de capabilité et avec les même tests, par contre l'heuristique basée sur la corrélation n'assure que 79.14%.

L'ensemble minimal de tests assurant une couverture de 100% des circuits défectueux est $\{5, 3, 1, 4\}$. Donc la méthode d'ordonnancement des tests a retrouvé 3 tests sur 4. La combinaison de l'ordre paramétrique composé des tests $\{3, 1, 5\}$ et l'ensemble minimal couvrant 100% des circuits défectueux $\{5, 3, 1, 4\}$ est illustré dans le Tableau 6.13.

Nombre de tests	**Séquence de tests**	**FC (%) par test**	**FC (%) cumulé**	**Intervalle de confiance du taux de défauts paramétriques à 95%**
1	5	87.05	87.05	[151.52 ; 152.70]
2	3	76.98	97.84	[24.67 ; 25.99]
3	1	71.94	99.28	[0 ; 0]
4	4	26.62	100	[0 ; 0]

Tableau 6.19 – Ensemble compact et ordonné du LNA.

6.4 Résumé des résultats

Dans cette section, nous résumons l'ensemble des résultats obtenus par la méthode de sélection ainsi que la méthode de sélection et d'ordonnancement. L'ensemble minimal couvrant 100% des circuits défectueux pour chacune des 6 catégories est illustré dans le Tableau 6.20.

N°	**Catégorie**	**Type**	**Nombre de tests**	**Tests retenus**		**Tests éliminés**	
				nombre	%	**nombre**	%
1	Courant d'alimentation	DC	34	3	8.82	31	91.18
2	Convertisseur numérique/analogique (DAC)	Mixte	6	4	66.67	2	33.33
3	Boucle de verrouillage de phase (PLL)	Mixte	7	2	28.57	5	71.43
4	Filtre	RF	19	3	15.79	16	84.21
5	Mixer	RF	43	11	25.58	32	74.42
6	Amplificateur faible bruit (LNA)	RF	8	4	50	4	50
Total			117	**27**		**90**	
Pourcentage (%)				**23.08**		**76.92**	

Tableau 6.20 – Résumé de l'application de la méthode de sélection.

L'analyse des résultats montre que la méthode de sélection permet de réduire le nombre total de tests à 27 (23.08%), en éliminant 90 tests (76.92%) sur l'ensemble des 117 tests. La première catégorie de tests (courant d'alimentation) a vu le nombre de ses tests se réduire de 91.18%, en se limitant à tester uniquement 3 tests sur les 34 du départ. Quant au convertisseur numérique/analogique, le nombre de ses tests éliminés n'est que de 2 tests sur un total de 6 tests. Le nombre de tests de chacune des 6 catégories à été réduit, en construisant des ensembles de tests qui garantissent une couverture de 100% des circuits défectueux.

Les résultats de l'application de la méthode de sélection et d'ordonnancement sont résumés dans le Tableau 6.21.

N°	Catégorie	Type	Nombre de tests	Tests retenus		Tests éliminés	
				nombre	%	nombre	%
1	Courant d'alimentation	DC	34	4	11.76	30	88.24
2	Convertisseur numérique/analogique (DAC)	Mixte	6	4	66.67	2	33.33
3	Boucle de verrouillage de phase (PLL)	Mixte	7	2	28.67	5	71.43
4	Filtre	RF	19	5	26.32	14	73.68
5	Mixer	RF	43	23	53.49	20	46.51
6	Amplificateur faible bruit (LNA)	RF	8	4	50	4	50
Total			117	**42**		**75**	
Pourcentage (%)				**35.9**		**64.1**	

Tableau 6.21 – Résumé de l'application de la méthode de sélection et d'ordonnancement.

L'analyse des résultats montre que la méthode de sélection et d'ordonnancement permet de réduire le nombre de tests à 35.9%, en éliminant 64.1% des tests redondants. Les tests du courant d'alimentation ont été réduits de 88.24%, par le test de seulement 4 tests sur les 34 tests de cette catégorie. Par contre, la diminution du nombre de tests de la deuxième catégorie n'a été que de 2 test sur les 6 tests d'origine. Toutes les autres catégories de tests ont vu leur nombre de tests se réduire avec des taux intermédiaires entre les taux des deux premières catégories. Les ensembles de tests obtenus garantissent une couverture de 100% des circuits défectueux et une détection de toutes les fautes paramétriques.

6.5 Conclusion

Dans ce chapitre, on a pu appliquer avec succès la méthode d'ordonnancement des tests sur un circuit industriel avec un nombre important de tests. Les différents tests ont été traités par catégorie au lieu de les traiter en un seul ensemble car les tests sont spécifiques à chaque bloc.

L'utilisation des copules, après un test de normalité non validé pour les différentes catégories de tests, a permis de construire des modèles statistiques plus fiables et avec des temps de génération courts. L'ordonnancement des tests a pu être traité par la méthode exacte du branch and bound au lieu des méthodes heuristiques (algorithmes génétique et recherche flottante), malgré le nombre important de tests de certaines catégories de tests comme le mixer avec 43 tests et le filtre avec 19 tests. Ceci a été rendu possible par la constatation qu'un nombre important de tests n'avaient aucun effet sur l'ordre d'élimination (taux de défauts nul). Donc une élimination préalable de ces tests a permis de traiter les tests restants par la méthode exacte du branch and bound.

Malgré l'utilisation unique des données synthétiques correspondantes à des déviations paramétriques, dans la construction des différents ordres d'éliminations, on a pu valider les résultats sur des circuits défectueux expérimentaux. Sauf dans le cas du convertisseur numérique/analogique où les données présentent peu de variation paramétrique avec des

spécifications de test très larges. Dans ce cas, on préconise d'effectuer tous les 6 tests sans aucune élimination. Dans les autres catégories de tests l'ordre d'élimination a été validé dans le cas des circuits défectueux. Ainsi, l'application de la méthode de sélection a permis de réduire le nombre total des tests de 76.92%, et la méthode de sélection et d'ordonnancement de 64.1%.

Chapitre 7

Conclusions et perspectives

7.1 Conclusions

Le test des circuits analogiques, mixtes et RF nécessite la vérification de chacun des tests d'un circuit par rapport à ses spécifications fixées dans le cahier des charges. Le test permet de classer chaque circuit comme fonctionnel s'il vérifie toutes les spécifications ou défectueux si au moins une des spécifications est violée. Ce procédé est très coûteux en termes de temps et coût du test, d'où la nécessité d'une optimisation du test.

Dans ce travail, on a proposé une méthode d'ordonnancement qui permet de tester en premier les tests détectant plus de circuits défectueux. Avec cet ordre de tests, les circuits défectueux seront détectés au plus tôt, ce qui permettra de réduire le temps de test car des le premier test violé le circuit est rejeté et le test interrompu. Par contre, cet ordre n'a aucune incidence sur les circuits fonctionnels qui vont effectués tous les tests. Notre méthode offre aussi la possibilité de réduire le temps de test des circuits fonctionnels, en identifiant les tests redondants dont on peut se passer car leur comportement est contenu dans les tests restants.

La méthode d'ordonnancement est applicable sur des circuits avec peu de données, en utilisant la modélisation statistique. Différents modèles sont proposés : une modélisation par une loi multinormale ou une copule dans le cas paramétrique. Dans le cas ou aucun des deux modèles précédents ne peut modéliser le comportement du circuit sous test, on fait appel à un modèle plus général non paramétrique. Ce dernier est applicable à tous les circuits mais possède une précision inférieure aux deux modèles paramétriques. Une fois le modèle statistique validé, le rééchantillonnage du modèle permet de générer un échantillon de plusieurs millions d'instances du circuit sous test. Ce grand échantillon permettra une meilleure estimation du taux de défauts aux ppm prés, qui va servir aux algorithmes de recherche pour ordonner les tests.

Nous avons implémenté différents algorithmes de recherche en fonction du nombre de tests du circuit sous test. Avec un circuit comportant peu de tests, quelque dizaine, la méthode exacte de séparation et d'évaluation permet de donner l'ordre optimal des

tests. Par contre, avec un circuit composé de plusieurs centaines de tests, la méthode de séparation et d'évaluation devient inapplicable. Dans ce cas, on fait appel aux méthodes heuristiques : algorithmes génétiques et méthodes de recherche flottante pour approcher l'ordre optimal des tests.

Pour bénéficier de l'optimalité de l'ordre des tests donné par la méthode de séparation et d'évaluation, nous avons proposé la méthode de décomposition. Cette méthode décompose l'ensemble des tests d'un circuit complexe en sous-ensembles de petite taille. Une fois les tests des sous-ensembles ordonnés d'une manière exacte par la méthode de séparation et d'évaluation, les différents ordres des sous-ensembles sont combinés pour fournir l'ordre final de tous les tests du circuit sous test.

Suivant la nature des données disponibles, la méthode d'ordonnancement des tests à été généralisée pour aboutir à trois variantes de la méthode d'ordonnancement :

- la méthode d'ordonnancement des tests traite les circuits dont on ne dispose que de peu de données sur les circuits fonctionnels ;
- la méthode de sélection traite les circuits ayant des données sur les circuits défectueux ;
- la méthode de sélection et d'ordonnancement ordonne les tests d'un circuit dont les données comportent des circuits fonctionnels et défectueux.

La méthode d'ordonnancement des tests et ses variantes ont été appliquées avec succès sur des circuits avec des données issus de la simulation Monte Carlo ainsi que sur un circuit industriel. Les circuits avec des données issues de la simulation Monte Carlo comme l'amplificateur opérationnel avec 11 tests et le LNA avec 5 tests ont été traités et l'ordre des tests à été validé par des modèles de fautes catastrophiques. L'application de la méthode proposée dans ce travail à été validé sur un circuit industriel avec 117 tests. La méthode de sélection a permis de construire des ensembles de tests pour chacune des 6 catégories de tests, en garantissant la détection de 100% des circuits défectueux. L'application de la méthode de sélection a réduit l'ensemble des tests de 76.92%, en ne retenant que 23.08% sur le total des 117 tests. La méthode de sélection et d'ordonnancement construit des ensembles de tests détectant toutes les fautes paramétriques ainsi que 100% des circuits défectueux. Son application a éliminée 64.1% des tests et n'a retenue que 35.9% des tests du circuit.

7.2 Perspectives

La méthode d'ordonnancement des tests proposée dans ce travail permet d'ordonner les tests de tous les types de circuits. La généralité de la méthode est due essentiellement aux techniques de modélisation statistique employées pour modéliser le comportement des tests ainsi qu'aux différents algorithmes de recherche employés suivant la complexité du circuit sous test. En plus, la méthode d'ordonnancement a été améliorée pour la prise en

compte des circuits défectueux à travers la méthode de sélection et la méthode de sélection et d'ordonnancement.

Plusieurs extensions de la mise en ouvre de la méthode peuvent être envisagées dans les travaux futurs :

- rendre la méthode adaptatif ;
- une implémentation de la méthode d'ordonnancement des tests indépendante de Matlab ;
- optimisation des codes des différents modules ;
- adaptation de la méthode aux spécificités des testeurs (ATE) ;
- prise en charge du test multi-site sur les nouveaux testeurs (test de plusieurs circuits en même temps).

Malgré l'extension de la méthode pour le traitement des données issues des circuits défectueux, la méthode d'ordonnancement peut être améliorée pour tenir en compte du test adaptatif. Il s'agit d'impacter le modèle et les résultats de toutes nouvelles données obtenues au cours du test. En effet, au cours du test des circuits après la production, de nouvelles données sont obtenues. Ces dernières peuvent être utilisées pour améliorer le modèle statistique ainsi que les résultats de la méthode d'ordonnancement. La configuration actuelle permet juste de tenir en compte des nouvelles données quelles soient issues de circuits fonctionnels ou défectueux qu'à travers une nouvelle modélisation et résolution du problème. Ça serait intéressent que les nouvelles données soient directement incorporées dans le modèle et surtout accéléré la résolution. Il est primordial que le temps séparant l'acquisition des nouvelles données et la mise à jour de la solution soit le plus cours possible pour permettre une meilleure réactivité sur les opérations de test en cours.

L'environnement de programmation influe considérablement sur le temps de simulation de la méthode proposée. L'implémentation actuelle des toutes les méthodes proposés à été réalisée sur Matlab. Ce dernier offre plusieurs outils qui ont permis d'accélérer l'implémentation des codes par la réutilisation de modules déjà implémentés comme dans le cas de la bibliothèque statistique contenant les modèle utilisés sauf dans le cas du modèle non paramétrique. Un inconvénient de cette implémentation et la dépendance de Matlab, donc le code ne pas être exécuter sans Matlab. Un autre inconvénient de Matlab est la lourdeur d'exécution comparé aux modules développés directement en langage C. Le temps d'exécution des simulations sera accéléré en s'affranchissant de Matlab.

Dernièrement une nouvelle génération de testeur a vue le jour, il s'agit des testeurs multi-sites. Ces derniers accélèrent les opérations de test en effectuant du test parallèle de plusieurs circuits en même temps. Une mise en ouvre de notre méthode d'ordonnancement sur ses testeurs peut être envisagée et expérimentée.

Bibliographie

[1] P. Duhamel and J.C. Rault. Automatic Test Generation Techniques for Analog Circuits and Systems : A Review. *IEEE Trans. on Circuits and Systems-II : Analog and Digital Signal Processing*, 26(7) :411–440, July 1979.

[2] T.W. Williams. *VLSI testing.* Elsevier Science Publishers B. V., Amsterdam, The Netherlands, 1986.

[3] R.J. Harvey, A.M.D. Richardson, and H.G. Kerhoff. Defect Oriented Test Developement Based on Layout inductive Fault Analysis. In *IEEE International Mixed-Signal Testing Workshop*, pages 2–9, 1995.

[4] F. Azais. *Conception en vue du test pour Circuits Integrés Analogiques et Mixtes.* PhD thesis, université Montpellier II, France, 1996.

[5] M. Sachdev and J.P.d. Gyvez. *Defect-Oriented Testing for Nano-Metric CMOS VLSI Circuits 2nd Edition (Frontiers in Electronic Testing).* Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.

[6] N. Ben-Hamida, K. Saab, D. Marche, and B. Kaminska. A perturbation based fault modelling and simulation for mixed-signal circuits. In *IEEE Asian Test Symposium*, pages 182–187, 1997.

[7] M. Zwolinski, A.D. Brown, and C.D. Chalk. Concurrent Analog Fault Simulation. In *IEEE International Mixed-Signal Testing Workshop*, pages 42–47, 1997.

[8] L. Milor. A Tutorial Introduction to Research on Analog and Mixed-Signal Circuits Testing. *IEEE Trans. on Circuits and Systems-II : Analog and Digital signal Processing*, 45(10) :1389–1407, October 1998.

[9] R. Leveugle. Test des circuits intégrés numériques : Notions de base. Génération de vecteurs. *Techniques de l'ingénieur. Electronique*, vol. 2, noE2460 :E2460.1–E2460.14, Août 2002.

[10] S. Battacharya, A. Halder, G. Srinivasan, and A. Chatterjee. Alternate testing of RF transceivers using optimized test stimulus for accurate prediction of system specifications. *In Journal of Electronic Testing : Theory and Applications (JETTA)*, 21 :323–339, 2005.

[11] A. Halder, S. Battacharya, and A. Chatterjee. Automatic multitone alternate test generation for RF circuits using behavioural models. In *In International Test Conference (ITC'03)*, pages 665–673, 2003.

[12] A. Halder, S. Bhattacharya, and A. Chatterjee. Alternate Test Generation for Specification Test of RF Circuits. In *IEEE International Mixed-Signal Testing Workshop, Seville, Spain,*, pages 7–12, Juin 2003.

[13] G. Srinivasan, A. Halder, S. Battacharya, S. Goyal, and A. Chatterjee. Loopback test of RF transceivers using periodic bit sequences : An alternate test approach. In *In International Mixed-Signal Test Workshop (IMSTW'04)*, pages 82–87, Juin 2004.

[14] T.W. Williams and N.C. Brown. Defect Level as a Function of Fault Coverage. *IEEE Transactions on Computers*, 30(12) :987–988, 1981.

[15] J.T. De Sousa, F.M. Goncalves, J.P. Teixeira, C. Marzocca, F. Corsi, and T.W.; Williams. Defect level evaluation in an IC design environment. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 15(10) :1286–1293, oct 1996.

[16] S. Sunter and N. Nagi. Test metrics for analog parametric faults. In *17th IEEE VLSI Test Symposium*, pages 226–234, 1999.

[17] A. Bounceur, S. Mir, E. Simeu, and L. Rolíndez. On the accurate estimation of test metrics for multiple analogue parametric deviations. In *12th IEEE International Mixed Signals Testing Workshop*, pages 19–26, Edinburgh, UK, June 21-23, 2006.

[18] S.D. Huss and R.S. Gyurcsik. Optimal ordering of analog integrated circuit tests to minimize test time. In *Design Automation Conference, 1991. 28th ACM/IEEE*, pages 494 –499, 1991.

[19] L. Milor and A.L. Sangiovanni-Vincentelli. Minimizing Production Test Time to detect Faults in Analog Circuits. *IEEE Trans. Computer-Aided Design*, 13(6) :796–813, June 1994.

[20] T.M. Souders and G.N. Stenbakken. Cutting the high cost of testing. *Spectrum, IEEE*, 28(3) :48 –51, mar 1991.

[21] J.B. Brockman and S.W. Director. Predictive subset testing : Optimizing IC parametric performance testing for quality, cost, and yield. *IEEE Transactions on Semiconductor Manufacturing*, 2(3) :104–113, 1989.

[22] D. Han, A. Halder, and A. Chatterjee. Test Elimination using Redundancy Analysis for Specification Test of Analog Circuits. In *10th IEEE International Mixed Signal Testing Workshop*, pages 69–75, Portland, June 23-25 2004.

[23] W. Jiang and B. Vinnakota. Defect-oriented test scheduling. *IEEE Transactions on VLSI Systems*, 9(3) :427–438, 2001.

[24] N. Akkouche, A. Bounceur, S. Mir, and E. Simeu. Minimization of functional tests by statistical modelling of analogue circuits. In *Proc. DTIS Design & Technology of Integrated Systems in Nanoscale Era International Conference on*, pages 35–40, 2007.

[25] C.-Y. Chao, H.-J. Lin, and L. Milor. Optimal testing of VLSI analog circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,*, 16(1) :58–77, January 1997.

[26] E. Acar and S. Ozev. Parametric test development for RF circuits targeting physical fault locations and using specification-based fault definitions. In *Proc. of ICCAD*, pages 73–79, 2005.

[27] A. Bounceur, S. Mir, E. Simeu, and L. Rolíndez. Estimation of Test Metrics for the Optimisation of Analogue Circuit Testing. *Journal of Electronic Testing : Theory and Applications*, 23 :471–484, 2007.

[28] C. Wegener. *Application of Linear Modeling to Testing and Characterizing D/A and A/D Converters*. PhD thesis, National University of Ireland, Cork, November 2003.

[29] A. Wrixon and M.P. Kennedy. A rigorous exposition of the LEMMA method for analog and mixed-signal testing. *IEEE Transactions on Instrumentation and Measurement*, 48(5) :978–985, October 1999.

[30] R.A. Horn and C.R. Johnson. *Topics in Matrix Analysis*. Cambridge university Press, 1991.

[31] T. Söderström and P. Stoica. *System Identification*. Prentice-Hall, Englewood Cliffs, NJ, 1989.

[32] T.M. Souders and G.N. Stenbakken. A Comprehensive Approach for Modeling and Testing Analog and Mixed-Signal Devices. *International Test Conference*, 7(1) :169–176, 1990.

[33] T.M. Souders and G.N. Stenbakken. Modeling and Test Point Selection for Data Converter Testing. In *International Test Conference (Philadelphia, PA)*, pages 813–817, 1985.

[34] H.-G. Stratigopoulos, P. Drineas, M. Slamani, and Y. Makris. Non-RF to RF test correlation using learning machines : A case study. In *Proc. VTS*, pages 9–14, May 2007.

[35] H.-G. Stratigopoulos, P. Drineas, M. Slamani, and Y. Makris. RF Specification Test Compaction Using Learning Machines. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 18(6) :998 –1002, june 2010.

[36] S. Biswas, P. Li, R.D. Blanton, and L.T. Pileggi. Specification test compaction for analog circuits and MEMS. In *Proceedings of the Design, Automation and Test in Europe, (DATE'05)*, pages 164–169, March 2005.

[37] S. Biswas and R.D. Blanton. Statistical Test Compaction Using Binary Decision Trees. *IEEE Design & Test of Computers*, 23(6) :452–462, June 2006.

[38] V.N. Vapnik. *Statistical learning theory*. Wiley-Interscience Publisher, New York, NY, 1998.

[39] Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, USA, 1 edition, January 1996.

[40] H.-G. Stratigopoulos and Y. Makris. Error Moderation in Low-Cost Machine-Learning-Based Analog/RF Testing. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 27(2) :339 –351, feb. 2008.

[41] B. Atzema and T. Zwemstra. Exploit analog IFA to improve specification based tests. In *European Design and Test Conference, 1996. ED TC 96. Proceedings*, pages 542–546, 11-14 1996.

[42] C.-Y. Pan and K.-T. Cheng. Pseudorandom testing for mixed-signal circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 16(10) :1173 –1185, oct 1997.

[43] W.M. Lindermeir, H.E. Graeb, and K.J. Antreich. Analog testing by characteristic observation inference. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 18(9) :1353 –1368, sep 1999.

[44] S. Krishnan and A. Zjajo. An Industrial to Re-order Integrated Circuit Tests to Maximize Test Benefit. In *IEE European Test Symposium, United Kingdom*, pages 253–258, May 2006.

[45] Roger B. Nelsen. *An Introduction to Copulas (Springer Series in Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[46] B.W. Silverman. *Density Estimation for Statisics and Data Analysis*. Chapman & Hall/CRC, 1986.

[47] M. Tan, H-B. Fang, G-L. Tian, and G. Wei. Testing multivariate normality in incomplete data of small sample size. *Journal of Multivariate Analysis*, 93(1) :164–179, 2005.

[48] A. Von Eye and G.A. Bogat. Testing the assumption of multivariate normality. *Psychology Science*, 46(2) :243–258., 2004.

[49] Z. Li-Xing, L.W. Hoi, and F. Kai-Tai. A test for multivariate normality based on sample entropy and projection pursuit. *Journal of Statistical Planning and Inference*, 45(3) :373–385, June 1995.

[50] J.F. Malkovich and A.A. Afifi. On tests for multivariate normality. *Journal of the american statistical association*, 68(341) :176–179, March 1973.

[51] C.J. Mecklin and D.J. Mundfrom. Comparing the Power of Classical and Newer Tests of Multivariate Normality. Technical report, Annual Meeting of the American Educational Research Association (81st)., New Orleans, LA, April 24-28 2000.

[52] R. Wehrens, H. Putter, and L.M.C. Buydens. The bootstrap : a tutorial . *Chemometrics and Intelligent Laboratory Systems*, 45(54) :35–52, August 2000.

[53] R. Palm. Utilisation du bootstrap pour les problèmes statistiques liés à l'estimation des paramètres. *Biotechnol. Agron. Soc. Environ*, 3(6) :143–153, August 2002.

[54] P. Hominal and P. Deheuvels. Estimation non paramétrique de la densité compte-tenu d'information sur le support. *Revue de statistique appliqué*, 27(3) :47–68, 1979.

[55] P. Deheuvels. Estimation non paramétrique de la densité par histogrammes généralisés. *Revue de statistique appliqué*, 3(25) :5–42, 1977.

[56] J. Shao. *Mathematical Statistics*. Springer-Verlag, New York, 1999.

[57] A. Ihler. Kernel Density Estimation Toolbox for Matlab. `http://www.ics.uci.edu/~ihler/code/kde.html`, Accessed March 2008.

[58] U. Cherubini, E. Luciano, and W. Vecchiato. *Copula methods in finance*. Wiley finance series. Wiley, Chichester [u.a.], 2004.

[59] A. Bounceur and S. Mir. Estimation of test metrics for AMS/RF BIST using Copulas. In *14th IEEE International Mixed-Signals, Sensors and Systems Test Workhop*, Vancouver, Canada, June 2008.

[60] A. Sklar. *Fonctions de Répartition a n Dimensions et Leurs Marges)*. Publications de l'Institut Statistique de l'Université de Paris 8, 1959.

[61] Y. Malevergne and D. Sornette. Testing the Gaussian Copula Hypothesis for Financial Assets Dependences. Finance 0111003, EconWPA, November 2001.

[62] M. Burns and G.W. Roberts. *An Introduction to Mixed-Signal IC Test and Measurement*. Test Economics, 2001.

[63] H. S. Wilf. *Algorithms and Complexity*. A. K. Peters, Ltd., Natick, MA, USA, 2nd edition, 2002.

[64] P.M. Narendra and K. Fukunaga. A Branch and Bound Algorithm for Feature Subset Selection. *IEEE Transaction on Computer*, 26(9) :917–922, August 1977.

[65] S.D. Stearns. On selcting features for pattern classifiers. In *third int. Conf. on Pattern recognition, Coronado, CA*, pages 71–75, 1976.

[66] S.S. Akbay and A. Chatterjee. Fault-based alternate test of RF components. In *Proc. 25th International Conference on Computer Design ICCD 2007*, pages 518–525, October 2007.

[67] N. Akkouche. Techniques d´optimisation du test analogique en utilisant des méthodes statistiques . In *Master Thesis, Mathématiques appliquées*, Université Jean Monnet de Saint Etienne, France, Juin 2006.

[68] N. Akkouche. Optimization of production test of analog and RF circuits using statistical modeling techniques. In *PhD Forum at IEEE Design, Automation and Test in Europe Conference, Grenoble*, March 2011.

[69] N. Akkouche, A. Bounceur, and S. Mir. Réduction de tests fonctionnels par modélisation statistique des circuits analogiques . In $10^{ième}$ *Journées Nationales du Réseau Doctoral en Micro-électronique*, Lille, France, Mai 2007.

[70] N. Akkouche, A. Bounceur, S. Mir, and E. Simeu. Functional test compaction by statistical modelling of analogue circuits. In 13^{th} *IEEE International Mixed-Signals Testing Workshop*, pages 20–24, Porto, Portugal, June 2007.

[71] N. Akkouche, S. Mir, and E. Simeu. Ordering of analog specification tests based on parametric defect level estimation. In *28th IEEE VLSI Test Symposium*, pages 301–306, Santa Cruz, USA, April 2010.

[72] N. Akkouche, S. Mir, E. Simeu, and H.-G. Stratigopoulos. Réduction de tests fonctionnels en utilisant des techniques d'estimation non paramétrique. In $11^{ième}$ *Journées Nationales du Réseau Doctoral en Micro-électronique*, Bordeaux, May 2008.

[73] P. Banerjee and J.A. Abraham. Fault characterization of VLSI MOS circuits. In *IEEE int. Conf. Circuits and Computers*, pages 564–568, 1982.

[74] B.R. Epstein, M. Czigler, and S.R. Miller. Fault detection and classification in linear integrated circuits : an application of discrimination analysis and hypothesis testing. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 12(1) :102–113, 1993.

[75] A.V. Gomes and A. Chatterjee. Distance constrained dimensionality reduction for parametric fault test generator. In *Proc. of ATS*, pages 411–416, 2001.

[76] L. Milor and A.L.S. Vincentelli. Detection of Catastrophic Faults in Analog Integrated Circuits. *IEEE Trans. Computer-Aided Design*, 8(2) :114–130, February 1989.

[77] C.-Y. Pan and K.-T. Cheng. Test Generation for Linear Time-Invariant Analog Circuits. *IEEE Trans. on Circuits and Systems-II : Analog and Digital Signal Processing*, 46(5) :554–564, 1999.

[78] R Development Core Team. *R : A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria, 2005. ISBN 3-900051-07-0.

[79] L. Rolíndez, S. Mir, and Carbonéro J.-L. Design of a 96-dB audio $\Sigma\Delta$ ADC including a BIST technique for SNDR testing. In *21st Conference on Design of Circuits and Integrated Systems (DCIS'06)*, Barcelona, Spain, November 2006.

[80] R.E. Smith, D.E. Goldberg, and J.A. Earickson. SGA-C : A C-language Implementation of a Simple Genetic Algorithm, 1991.

[81] P. Somol, P. Pudil, J. Novovicova, and P. Paclik. Adaptive floating search methods in feature selection. *Pattern Recognition Letters*, 20 :1157–1163, 1999.

[82] H.-G. Stratigopoulos, S. Mir, E. Acar, and S. Ozev. Defect Filter for Alternate RF Test. In *Proc. 14th IEEE European Test Symposium*, pages 101–106, May 2009.

[83] H.-G. Stratigopoulos, S. Mir, and A. Bounceur. Evaluation of Analog/RF Test Measurements at the Design Stage. *IEEE Trans. Computer-Aided Design*, 28(4) :582–590, April 2009.

[84] Z. Wang, G. Gielen, and W. Sansen. Probabilistic fault detection and the selection of measurements for analog integrated circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 17(9) :862 –872, sep 1998.

[85] E. Wegman. Non parametric density estimation I, A summary of avaible methods. *Technometrics*, 27(14) :533–546, 1972.

[86] W. Wertz. *Statistical Density Estimation*. Vandenhoeck & Ruprecht, 1978.

Liste des publications de l'auteur

Conférences internationales et Workshops avec comité de lecture

- [1] **N. Akkouche**, S. Mir, E. Simeu and M. Slamani. Analog/RF test ordering in the early stages of production testing, *30th IEEE VLSI Test Symposium (VTS'12)*, April 2012, Hyatt Maui, Hawaii, USA. (submitted)
- [2] **N. Akkouche**, Optimization of production test of analog and RF circuits using statistical modeling techniques, *PhD Forum at IEEE Design, Automation and Test in Europe Conference*, March 2011, Grenoble, France.
- [3] **N. Akkouche**, S. Mir and E. Simeu. Ordering of Functional Tests Based on Parametric Defect Level Estimation, *28th IEEE VLSI Test Symposium (VTS'10)*, April 2010, Santa Cruz, California, USA, pp. 301-306.
- [4] **N. Akkouche**, A. Bounceur, S. Mir and E. Simeu. Minimization of functional tests by statistical modelling of analogue circuits. *In Design and Technology of Integrated Systems (DTIS)*, September 2007, Rabat, Morocco, pp. 35-40.
- [5] **N. Akkouche**, A. Bounceur, S. Mir and E. Simeu. Functional test compaction by statistical modelling of analogue circuits. *13th IEEE International Mixed-Signals Testing Workhop*, June 2007, Porto, Portugal, pp. 20-24.

Conférences nationales

- [6] **N. Akkouche**, S. Mir et E. Simeu, Modélisation statistique de circuits analogiques et mixtes pour l'optimisation du test de production, *Premières journées du projet SEmba*, Octobre 2009, Annecy, France.
- [7] **N. Akkouche**, S. Mir, E. Simeu and H. Stratigopoulos. Réduction de tests fonctionnels en utilisant des techniques d'estimation non paramétrique. $11^{ième}$ *Journées Nationales du Réseau Doctoral en Micro-électronique*, May 2008, Bordeaux, France.
- [8] **N. Akkouche**, A. Bounceur et S. Mir. Réduction de tests fonctionnels par modélisation statistique des circuits analogiques. $10^{ième}$ *Journées Nationales du Réseau Doctoral de Micro-électronique*, May 2007, Lille, France.

Optimisation of the production test of analog and RF circuit using statistical modeling techniques

Abstract : The share of test in the cost of design and manufacture of integrated circuits continues to grow, hence the need to optimize this step. In this thesis, new methods of test scheduling and reducing the number of tests are proposed. The solution is a sequence of tests for early identification of faulty circuits, which can also be used to eliminate redundant tests. These test methods are based on statistical modeling of the circuit under test. This model included several parametric and non-parametric models to adapt to all types of circuit. Once the model is validated, the suggested test methods generate a large sample containing defective circuits. These allow a better estimation of test metrics, particularly the defect level. Based on this error, a test scheduling is constructed by maximizing the detection of faulty circuits. With few tests, the Branch and Bound method is used to obtain the optimal order of tests. However, with circuits containing a large number of tests, heuristics such as decomposition method, genetic algorithms or floating search methods are used to approach the optimal solution.
Key words : analog and RF circuit, functional test, parametric faults, statistical modeling, test metrics, Feature Selection Algorithm.

Optimisation du test de production de circuits analogiques et RF par des techniques de modélisation statistique

Résumé : La part du test dans le coût de conception et de fabrication des circuits intégrés ne cesse de croître, d'où la nécessité d'optimiser cette étape devenue incontournable. Dans cette thèse, de nouvelles méthodes d'ordonnancement et de réduction du nombre de tests à effectuer sont proposées. La solution est un ordre des tests permettant de détecter au plus tôt les circuits défectueux, qui pourra aussi être utilisé pour éliminer les tests redondants. Ces méthodes de test sont basées sur la modélisation statistique du circuit sous test. Cette modélisation inclus plusieurs modèles paramétriques et non paramétrique permettant de s'adapter à tous les types de circuits. Une fois le modèle validé, les méthodes de test proposées génèrent un grand échantillon contenant des circuits défectueux. Ces derniers permettent une meilleure estimation des métriques de test, en particulier le taux de défauts. Sur la base de cette erreur, un ordonnancement des tests est construit en maximisant la détection des circuits défectueux au plus tôt. Avec peu de tests, la méthode de sélection et d'évaluation est utilisée pour obtenir l'ordre optimal des tests. Toutefois, avec des circuits contenant un grand nombre de tests, des heuristiques comme la méthode de décomposition, les algorithmes génétiques ou les méthodes de la recherche flottante sont utilisées pour approcher la solution optimale.
Mots clés : circuit analogique et RF, test fonctionnel, fautes paramétriques, modélisation statistique, métriques de test, algorithme de recherche.

ISBN 978-2-84813-171-9