

THESE

Présentée pour obtenir le grade de **DOCTEUR**

Spécialité : DISPOSITIFS DE L'ELECTRONIQUE INTEGREE

par Hubert FILIOL

Ingénieur CPE Lyon

Master Recherche SIDS-EEA « Dispositifs de l'Electronique Intégrée »

Méthodes d'analyse de la variabilité et de conception robuste des circuits analogiques dans les technologies CMOS avancées

Soutenue le 22 juillet 2010 à Grenoble devant le Jury composé de :

G. Scorletti	Professeur, Ecole Centrale de Lyon	Président
G. Gielen	Professeur, Katholieke Universiteit Leuven	Rapporteur
Y. Hervé	Maître de Conférences, Université Louis Pasteur de Strasbourg	Rapporteur
I. O'Connor	Professeur, Ecole Centrale de Lyon	Directeur de thèse
D. Morche	Ingénieur, CEA-LETI, Grenoble	Co-Directeur de thèse
E. Remond	Ingénieur, STMicroelectronics, Crolles	Examineur
X. Jonsson	Ingénieur, Mentor Graphics, Montbonnot Saint-Martin	Examineur

« Le second [principe est] de diviser chacune des difficultés que j'examinerais en autant de parcelles qu'il se pourrait, et qu'il serait requis pour les mieux résoudre. »

René DESCARTES
Discours de la méthode

Remerciements

Les travaux de thèse présentés dans ce manuscrit ont été effectués dans le cadre d’une collaboration entre l’Institut des Nanotechnologies de Lyon (INL), UMR CNRS 5512 de l’École Centrale de Lyon et le Laboratoire d’Électronique et de Technologies de l’Information de Grenoble (CEA-LETI).

Je remercie Monsieur Guy Hollinger, en tant que directeur de l’INL, pour m’avoir donné l’opportunité d’effectuer cette thèse. Je remercie également Monsieur Jean-René Lequepeys, responsable du Service Conception pour les Microtechnologies Emergentes (SCME) au CEA-LETI et Monsieur Cyril Condemine, responsable du Laboratoire Microsystèmes et Electronique Associée (LMEA) au SCME, pour m’avoir accueilli au sein de leur entité et fourni les moyens nécessaires à la réalisation de mes travaux.

Je tiens tout particulièrement à remercier mon directeur de thèse Monsieur Ian O’Connor, pour son soutien et la confiance qu’il m’a accordée, ainsi que mon encadrant de thèse Monsieur Dominique Morche pour ses conseils et son aide précieuse.

Pour le temps et l’intérêt porté à l’examen de ce manuscrit, j’adresse ma reconnaissance à Monsieur Gérard Scorletti (Professeur à l’Ecole Centrale de Lyon), président ; Messieurs Georges Gielen (Professeur à la « Katholieke Universiteit Leuven ») et Yannick Hervé (Maître de Conférences à l’Université Louis Pasteur de Strasbourg), rapporteurs ; Messieurs Eric Remond (Ingénieur chez STMicroelectronics) et Xavier Jonsson (Ingénieur chez Mentor Graphics), examinateurs.

Je remercie par ailleurs tous les membres de l’équipe conception à l’INL et du laboratoire MEA au CEA-LETI, pour leur bonne humeur et leur enthousiasme au cours de ces trois années de thèse. Un grand merci également à tous mes collègues thésards, DRT ou stagiaires avec qui j’ai sympathisé.

Je tiens enfin à exprimer toute ma reconnaissance à mes proches pour leur soutien durant cette période et plus particulièrement Laetitia dont la patience et les encouragements ont été de vraies sources de motivation.

Résumé

Avec la miniaturisation toujours plus poussée des technologies CMOS, il devient de plus en plus difficile de maîtriser les variations des paramètres technologiques lors de la fabrication des circuits intégrés. A cause de ces variations, les performances des circuits peuvent varier de façon considérable. Par conséquent, des méthodes d'analyse de la variabilité et de conception robuste sont plus que jamais nécessaires pour garantir un rendement de fabrication des circuits élevé.

Les techniques classiques d'analyse de la variabilité se révèlent soit pessimistes conduisant alors à un surdimensionnement (analyse « pire-cas »), soit très couteuses en temps de calcul (analyse Monte Carlo). Quant aux méthodes de conception automatisée robuste, elles sont généralement basées sur des algorithmes d'optimisation locaux qui améliorent la robustesse des circuits localement, mais risquent de ne pas converger vers le dimensionnement globalement robuste.

Dans ce travail de thèse, une nouvelle méthode d'analyse de la variabilité ainsi qu'une nouvelle approche pour concevoir des circuits analogiques robustes ont été développées. La méthode d'analyse de la variabilité consiste à approximer les performances des circuits par des modèles polynomiaux à partir des plans d'expériences, puis à estimer les variations extrêmes grâce au développement limité de Cornish-Fisher. Cette méthode s'avère aussi précise que l'analyse de Monte Carlo, mais présente un coût calculatoire bien plus faible. Enfin, l'approche de conception robuste met en œuvre la méthode précédente d'analyse de la variabilité dans un algorithme d'optimisation par intervalles afin d'assurer un dimensionnement globalement robuste.

Summary

With the continuous downscaling of CMOS technology, precise control over process parameters has become a highly challenging task. Due to the fluctuations in the manufacturing process, the performance of integrated circuits (ICs) will vary greatly between chips. Therefore, efficient methods to analyze such variability are essential to guarantee that fabricated ICs will meet the design and yield specifications.

The classical methods for variability analysis are either pessimistic, thus leading to overdesign (worst-case analysis), or computing time expensive (Monte Carlo analysis). As for robust design methods, they are generally based on local optimization algorithms that locally improve the yield, but may not guarantee that the globally robust sizing is found.

In this work, a new method for variability analysis and a new approach to design robust analog circuits are developed. The method dedicated to variability analysis consists of building polynomial models of the circuit performance metrics with the Design of Experiments theory, and then estimating the extreme variations by means of the Cornish-Fisher expansion. Compared to Monte Carlo analysis, this method shows a good accuracy without the shortcoming of a large computing cost. Finally, the robust design approach applies the previous variability analysis method in an interval-based optimization algorithm to obtain a globally robust sizing.

Abréviations

CAO	Conception Assistée par Ordinateur
CMOS	Complementary Metal Oxide Semiconductor
IP	Intellectual Property
MC	Monte Carlo
MEMS	Micro-Electro-Mechanical Systems
MOS	Metal Oxide Semiconductor
MOSFET	Metal Oxide Semiconductor Field-Effect Transistor
NBTI	Negative Bias Temperature Instability
NEMS	Nano-Electro-Mechanical Systems
OTA	Amplificateur Opérationnel à Transconductance

Notations

X	Variable
$\mathbf{X}$	Vecteur ou matrice de variables
$\text{tr}(\mathbf{X})$	Trace de la matrice $\mathbf{X}$
$\mathbf{X}^T$	Transposée de la matrice $\mathbf{X}$
$[\mathbf{X}]$	Vecteur d'intervalles ou pavé
j	Nombre imaginaire
C_n^k	Combinaison de k éléments parmi n
$P(X > b)$	Probabilité que la variable aléatoire X soit supérieure à b
$E(X)$	Espérance d'une variable aléatoire X
$V(X)$	Variance d'une variable aléatoire X
$\text{cov}(X, Y)$	Covariance de deux variables aléatoires X et Y
$f_X(X)$	Densité de probabilité d'une variable aléatoire X
$F_X(X)$	Fonction de répartition d'une variable aléatoire X
$N(\mu, \sigma)$	Loi normale de moyenne μ et variance σ
$\hat{Y}$	Méta-modèle approximant une grandeur Y

Table des matières

Chapitre 1 : Introduction.....	1
1.1 Le contexte	1
1.2 Plan du mémoire.....	2
Chapitre 2 : Les limites des technologies CMOS nanométriques	3
2.1 La miniaturisation des technologies CMOS.....	3
2.1.1 Historique du transistor MOS	3
2.1.2 Loi de Moore	3
2.1.3 Règles de réduction d'échelle de Dennard	4
2.2 Les limites de la miniaturisation	6
2.2.1 Limites physiques : la barrière de l'atome.....	6
2.2.2 Méthodologies de conception : la maîtrise de la complexité.....	10
2.2.3 Coût des investissements R&D	11
2.3 Les phénomènes de variabilité	12
2.3.1 Sources des variations.....	13
2.3.2 Classification des variations paramétriques.....	13
2.3.3 Modélisation des variations paramétriques	15
2.3.4 Evolution de la variabilité avec la miniaturisation	17
2.4 Conclusion.....	17
Chapitre 3 : Etat de l'art des méthodes d'analyse de la variabilité et de conception robuste des circuits analogiques	19
3.1 Définitions.....	19
3.1.1 Simulation des circuits analogiques.....	19
3.1.2 Domaine de conception et régions de tolérance	21
3.1.3 Régions d'acceptabilité.....	22
3.1.4 Rendement paramétrique	23
3.2 Etat de l'art - Analyse de la variabilité.....	25
3.2.1 Problématique	25
3.2.2 Analyse pire-cas.....	26
3.2.3 Analyse analytique.....	27

3.2.4	Analyse de Monte Carlo	28
3.2.5	Estimation de la distribution des performances	31
3.2.6	Bilan.....	34
3.3	Etat de l’art - Méthodes de conception robuste.....	36
3.3.1	Problématique	36
3.3.2	Circuits discrets	37
3.3.3	Optimisation du rendement.....	38
3.3.4	Optimisation robuste - Optimisation des performances pire-cas.....	42
3.3.5	Optimisation par les indices de capabilité	44
3.3.6	Bilan.....	45
3.4	Problématique de la thèse.....	46

Chapitre 4 : Analyse de la variabilité à partir des modèles polynomiaux et de l’approximation de Cornish-Fisher 49

4.1	Présentation de la méthode d’analyse de la variabilité	49
4.2	Construction d’un modèle polynomial des performances avec les plans d’expériences	50
4.2.1	Intérêt des méta-modèles	50
4.2.2	Les plans d’expériences	51
4.2.3	Modèles polynomiaux d’ordre 1 et 2.....	59
4.2.4	Approximation des performances par des polynômes.....	72
4.3	Estimation des bornes de variation des performances	73
4.3.1	Bornes de variation des performances – Valeurs pire-cas.....	73
4.3.2	Modèle linéaire sans interactions.....	74
4.3.3	Modèle linéaire avec interactions et modèle quadratique – Développement limité de Cornish-Fisher	75
4.4	Application de l’analyse de variabilité aux circuits analogiques	81
4.4.1	Comparaison de méthodes.....	81
4.4.2	Porte OU exclusif	83
4.4.3	Amplificateur opérationnel à transconductance	85
4.4.4	Amplificateur opérationnel de Miller	87
4.4.5	Bilan.....	90
4.5	Conclusion.....	90

Chapitre 5 : Modélisation par morceaux des performances des circuits analogiques 93

5.1	Introduction	93
-----	--------------------	----

5.2	Méthode de modélisation par morceaux des performances	95
5.2.1	Algorithme de modélisation par morceaux.....	95
5.2.2	Modélisation séquentielle	96
5.2.3	Validation du modèle.....	97
5.2.4	Stratégie de bisection	97
5.2.5	Sauvegarde des points.....	100
5.2.6	Exemple numérique	101
5.3	Application à la modélisation des performances d'un amplificateur opérationnel à transconductance	103
5.3.1	Comparaison de méthodes	103
5.3.2	Variables abstraites	104
5.3.3	Résultats.....	106
5.4	Conclusion.....	108
Chapitre 6 : Conception robuste des circuits analogiques		111
6.1	Présentation de la méthode de conception robuste.....	111
6.2	Conception robuste des circuits analogiques	112
6.2.1	Robustesse d'un circuit.....	113
6.2.2	Modélisation des performances pire-cas.....	113
6.2.3	Formulation sous la forme d'un problème d'optimisation robuste	115
6.3	Algorithmes d'optimisation globale par intervalles	116
6.3.1	Optimisation globale avec contraintes	116
6.3.2	Algorithme par intervalles de base	117
6.3.3	Algorithmes par intervalles avec contracteurs.....	124
6.3.4	Comparaison des algorithmes d'optimisation par intervalles.....	136
6.4	Algorithmes par intervalles adaptés à la conception robuste des circuits analogiques	138
6.4.1	Conception robuste inspirée des algorithmes par intervalles	139
6.4.2	Bilan.....	142
6.5	Application aux circuits analogiques	143
6.5.1	Conception robuste d'un amplificateur opérationnel à transconductance	143
6.5.2	Conception robuste d'un amplificateur opérationnel de Miller.....	150
6.6	Conclusion.....	153
Chapitre 7 : Conclusion et perspectives		155
7.1	Analyse de la variabilité par les plans d'expériences et le développement limité de Cornish-Fisher.....	155
7.2	Modélisation par morceaux des performances.....	155

7.3 Optimisation robuste par intervalles	156
7.4 Perspectives	156
Bibliographie.....	159
Publications	171
Annexe A : Eléments de probabilité.....	173
1. Variable aléatoire.....	173
2. Vecteur de variables aléatoires	176
3. Loi normale.....	177
4. Loi du Khi-2 non centrée.....	178
Annexe B : Problème d’optimisation	179
Annexe C : Arithmétique par intervalles.....	181
1. Définitions	181
2. Surestimation	183
Annexe D : Implémentation en Java	185

Liste des figures

Figure 2.1 : Loi de Moore (source Intel)	4
Figure 2.2 : Evolution de différentes caractéristiques des technologies CMOS sur substrat d'après le rapport « Process Integration, Devices & Structures » de l'ITRS 2009	10
Figure 2.3 : Répartition spatiale des variations paramétriques et pourcentage de la variabilité totale.....	15
Figure 2.4 : Evolution de la variabilité de différents paramètres dans les futures technologies CMOS d'après l'ITRS.....	17
Figure 3.1 : Représentation de la simulation électrique des circuits analogiques	21
Figure 3.2 : Région tolérance $RT_{\Delta T}$ définie pour deux variations technologiques ΔT_1 et ΔT_2	22
Figure 3.3 : Régions d'acceptabilité A_Y et $A_{AT}(C, E)$ pour deux performances et deux variations technologiques.....	23
Figure 3.4 : Histogramme de la distribution d'une performance pour plusieurs circuits fabriqués.....	24
Figure 3.5 : Variabilité des performances causée par les variations paramétriques	25
Figure 3.6 : Méthodes d'analyse de la variabilité des circuits analogiques	26
Figure 3.7 : Illustration de la surestimation causée par les modèles pire-cas	27
Figure 3.8 : Approximations polyédrique (a) et ellipsoïdale (b) de la région d'acceptabilité.....	31
Figure 3.9 : Illustration de la probabilité de défaillance étant donné une spécification $\bar{B}$ sur une performance Y	33
Figure 3.10 : Méthodes de conception robuste des circuits analogiques.....	37
Figure 3.11 : Régions d'acceptabilité A_Y et A_C pour deux performances et deux paramètres de conception dans le cas des circuits discrets.....	37
Figure 3.12 : Déplacement de la région de tolérance R_T avec la méthode des centres de gravité.....	39
Figure 3.13 : Maximisation du rendement (volume bleu) en centrant la densité de probabilité sur la région d'acceptabilité (rectangle rouge).....	40
Figure 3.14 : Centrage de la distribution des performances sur la région d'acceptabilité A_Y (a) et centrage de la distribution des variations technologiques sur la région d'acceptabilité $A_{AT}(C)$ (b).....	41
Figure 3.15 : Illustration des distances pire-cas (flèches en vert).....	42
Figure 3.16 : Illustration des indices C_p (a) et C_{pk} (b).....	44
Figure 4.1 : Méthode d'analyse de la variabilité proposée.....	50
Figure 4.2 : Représentation d'un modèle numérique	51
Figure 4.3 : Représentations matricielle (a) et graphique (b) d'un plan d'expériences comportant deux facteurs et deux essais	55
Figure 4.4 : Algorithme de validation croisée « leave-one-out »	58
Figure 4.5 : Nombre d'essais des plans de Plackett-Burman, des plans fractionnaires et des plans complets en fonction du nombre de facteurs étudiés.....	67

Figure 4.6 : Plan composite centré avec deux facteurs.....	68
Figure 4.7 : Plan composite centré avec $\alpha = 1$	70
Figure 4.8 : Emplacement des points supplémentaires (en vert) pour tester un modèle linéaire (a) ou quadratique (b).....	71
Figure 4.9 : Bornes de variation d'une performance	74
Figure 4.10 : Densité de probabilité de Y (a) et d'une loi normale centrée réduite (b).....	79
Figure 4.11 : Histogramme du courant de fuite.....	84
Figure 4.12 : Amplificateur opérationnel à transconductance.....	85
Figure 4.13 : Histogrammes de la fréquence de coupure (a) et du gain différentiel (b).....	86
Figure 4.14 : Erreur relative sur l'estimation des bornes (1 ^{er} et 99 ^{ème} centiles) de la fréquence de coupure et du gain différentiel	86
Figure 4.15 : Amplificateur opérationnel de Miller.....	87
Figure 4.16 : Histogrammes de la fréquence de coupure (a) et du gain différentiel (b).....	88
Figure 4.17 : Histogrammes de la marge de phase (a) et de la consommation de courant (b)	88
Figure 4.18 : Erreur relative sur l'estimation des bornes (1 ^{er} et 99 ^{ème} centiles) de la fréquence de coupure et du gain différentiel	89
Figure 4.19 : Erreur relative sur l'estimation des bornes (1 ^{er} et 99 ^{ème} centiles) de la marge de phase et de la consommation en courant.....	89
Figure 5.1 : Comparaison des modèles polynomiaux en fonction de la précision et du coût de calcul	94
Figure 5.2 : Principe de la modélisation par morceaux	94
Figure 5.3 : Algorithme de modélisation par morceaux des performances.....	96
Figure 5.4 : Représentation graphique de la fonction h (a) et de son approximation quadratique f (b)	99
Figure 5.5 : Représentation graphique des composantes g_1 (a) et g_2 (b) du gradient de l'approximation quadratique f	99
Figure 5.6 : Modèles par morceaux de h en découpant le domaine initial perpendiculairement à la direction de X_1 (a) et perpendiculairement à la direction de X_2 (b)	99
Figure 5.7 : Points du plan composite centré simulés sur le domaine initial (a) et réutilisés sur le sous-domaine de « gauche » (b).....	100
Figure 5.8 : Modèles construits à chaque itération de l'algorithme de modélisation par morceaux	102
Figure 5.9 : Amplificateur opérationnel à transconductance.....	103
Figure 5.10 : Comparaison des trois modélisations appliquées au gain différentiel	106
Figure 5.11 : Comparaison des trois modélisations appliquées à la fréquence de coupure pour deux valeurs différentes de $RMSE_{max}$: 7.5 kHz (a) et 5 kHz (b).....	107
Figure 6.1 : Principe de la méthode de conception robuste.....	112
Figure 6.2 : Comparaison d'une spécification $\bar{B}$ et de la performance pire-cas $F_Y^{-1}(\alpha)$ pour un rendement minimum α donné	113
Figure 6.3 : Principe de la modélisation des performances pire-cas	115
Figure 6.4 : Le sous-pavé qui est extrait de la liste L en priorité est celui qui présente la plus petite borne inférieure $\underline{F}$. Dans l'exemple ci-dessus, c'est donc le sous-pavé $[X_1]$ qui est extrait de L	119

Figure 6.5 : Cas de figures possibles lors de l'étude de la satisfaction des contraintes à partir des bornes de l'image de G_I sur $[X]$	120
Figure 6.6 : Algorithme d'optimisation par intervalles.....	123
Figure 6.7 : Projections de l'ensemble solution $S_\bullet$ sur les domaines de x_g et x_d	128
Figure 6.8 : Procédure de propagation-rétropropagation appliquée à la contrainte $x_g \bullet x_d = y$	129
Figure 6.9 : Phase de propagation appliquée à la contrainte $x_1.x_2 + 2x_3 = y$	129
Figure 6.10 : Phase de rétropropagation appliquée à la contrainte $x_1.x_2 + 2x_3 = y$	130
Figure 6.11 : Algorithme du contracteur de propagation-rétropropagation.....	131
Figure 6.12 : Représentation graphique des contraintes du problème CSP (6.25). L'ensemble solution de chaque contrainte est représenté en bleu clair.	133
Figure 6.13 : Les pavés obtenus avec le contracteur par propagation-rétropropagation (a) et le contracteur par programmation quadratique (b) sont modélisés par les rectangles en bleu foncé. L'ensemble solution du problème CSP (6.25) est représenté en bleu clair.....	134
Figure 6.14 : Algorithme d'optimisation par intervalles avec contracteur.....	136
Figure 6.15 : Représentation graphique de la fonction de coût et du domaine de faisabilité (en bleu clair) du problème d'optimisation (6.26). Les points A et B représentent les minima du problème.	136
Figure 6.16 : Largeur des pavés traités à chaque itération des algorithmes par intervalles.....	138
Figure 6.17 : Conception robuste basée sur l'algorithme par intervalles de base	142
Figure 6.18 : Amplificateur opérationnel à transconductance.....	144
Figure 6.19 : Distributions des performances de l'OTA au point de conception robuste	146
Figure 6.20 : Caractéristique $gm/I_D = f(I_D/(W/L))$ d'une technologie CMOS 65nm	148
Figure 6.21 : Amplificateur opérationnel de Miller.....	149
Figure 6.22 : Intégration de la méthode de dimensionnement gm/I_D au sein de l'algorithme de conception robuste par intervalles.....	151
Figure 6.23 : Extraction des valeurs des paramètres de conception à partir des rapports gm/I_D	152
Figure 7.1 : Quantile d'ordre α de la distribution de X	174
Figure 7.2 : Illustration de l'effet enveloppant.....	183
Figure 7.3 : Illustration de l'effet de la décorrélation des données	184
Figure 7.4 : Représentation UML simplifiée de la classe mère Parameter et des classes dérivées.....	185
Figure 7.5 : Représentation UML simplifiée de la classe mère Function et des classes dérivées	186
Figure 7.6 : Représentation UML simplifiée de la classe MetamodelingFactory et de ses interactions avec les autres classes	186
Figure 7.7 : Flot de la conception robuste	187

Liste des tableaux

Tableau 2.1 : Règles de réduction d'échelle du transistor MOS proposées par R. H. Dennard.....	5
Tableau 3.1 : Méthodes d'analyse de la variabilité.....	35
Tableau 3.2 : Méthodes de conception robuste	46
Tableau 4.1 : Comparaison de différents types de méta-modèles en fonction de la dimension (nombre de paramètres) et du degré de non-linéarité du modèle numérique à approximer d'après [90]	54
Tableau 4.2 : Matrice des essais d'un plan factoriel complet 2^3	61
Tableau 4.3 : Matrice du modèle linéaire complet	61
Tableau 4.4 : Matrice des essais d'un plan factoriel fractionnaire 2^{5-1}	63
Tableau 4.5 : Matrice du modèle linéaire avec interactions d'ordre 1	64
Tableau 4.6 : Matrice des essais d'un plan de Plackett-Burman avec sept facteurs	65
Tableau 4.7 : Matrice du modèle linéaire sans interactions avec sept facteurs	66
Tableau 4.8 : Coût calculatoire des plans factoriels en termes d'essais.....	67
Tableau 4.9 : Matrice des essais d'un plan composite centré à deux facteurs	69
Tableau 4.10 : Matrice du modèle quadratique avec deux facteurs.....	69
Tableau 4.11 : Comparaison des caractéristiques des modèles linéaires et quadratiques	72
Tableau 4.12 : Coût calculatoire des trois approches	83
Tableau 4.13 : Etude d'une porte OU exclusif.....	84
Tableau 4.14 : Etude d'un amplificateur opérationnel à transconductance.....	85
Tableau 4.15 : Temps de calcul pour construire les modèles et estimer les bornes de variation de l'amplificateur opérationnel à transconductance	86
Tableau 4.16 : Etude d'un amplificateur opérationnel de Miller.....	87
Tableau 4.17 : Temps de calcul pour construire les modèles et estimer les bornes de variation de l'amplificateur opérationnel de Miller.....	89
Tableau 5.1 : Modèle construit à chaque itération de l'algorithme de modélisation par morceaux.....	101
Tableau 5.2 : Paramètres de l'amplificateur opérationnel à transconductance.....	104
Tableau 5.3 : Variables abstraites.....	105
Tableau 5.4 : Temps de simulation pour construire les modèles.....	106
Tableau 6.1 : Comparaison des contracteurs C_{PR} et C_{PQ}	134
Tableau 6.2 : Performances des algorithmes par intervalles pour résoudre le problème (6.26).....	137
Tableau 6.3 : Paramètres de l'amplificateur opérationnel à transconductance.....	144
Tableau 6.4 : Variables d'optimisation	145
Tableau 6.5 : Comparaison du coût calculatoire des deux méthodes	145
Tableau 6.6 : Résultats de la conception robuste de l'OTA	146
Tableau 6.7 : Variations des paramètres technologiques de l'amplificateur opérationnel de Miller.....	152

Tableau 6.8 : Variables d'optimisation	152
Tableau 6.9 : Coût calculatoire de la conception robuste de l'amplificateur de Miller.....	153
Tableau 6.10 : Résultats de la conception robuste de l'amplificateur de Miller.....	153

Chapitre 1 : Introduction

1.1 Le contexte

La miniaturisation des composants microélectroniques se heurte à des phénomènes physiques de plus en plus contraignants à l'approche de l'échelle nanométrique. Parmi ces phénomènes, les dispersions paramétriques tiennent une place significative de par leur impact sur les performances des circuits intégrés. En effet, ces dispersions des paramètres technologiques, qui résultent des fluctuations lors du processus de fabrication, engendrent une certaine variabilité au niveau des performances des circuits. Or, si les performances sont trop dispersées, une certaine proportion des circuits fabriqués ne respectent pas les spécifications fixées par le cahier des charges, ce qui conduit à une dégradation du rendement de fabrication et donc une augmentation des coûts de production pour les fabricants de semi-conducteurs. Des méthodes d'analyse de la variabilité et de conception robuste des circuits sont donc indispensables afin de proposer des solutions à ces problèmes de dispersions.

Les méthodes d'analyse de la variabilité consistent généralement à évaluer une grandeur statistique qui caractérise cette variabilité (valeurs extrêmes, distribution des performances, etc.). Les deux méthodes classiques sont l'analyse pire-cas et l'analyse de Monte Carlo. L'analyse pire-cas a pour objectif de déterminer les valeurs extrêmes des performances en combinant les valeurs extrêmes des paramètres technologiques. Bien que très rapide, cette méthode est cependant imprécise dans le cas des circuits analogiques, conduisant à une estimation pessimiste des performances et donc un surdimensionnement. A l'opposé, l'analyse de Monte Carlo est d'une grande précision mais nécessite la simulation de milliers d'échantillons afin d'approximer la distribution des performances avec exactitude. De nouvelles méthodes d'analyse de variabilité, à la fois précises et efficaces en termes de coût calculatoire, sont donc nécessaires.

Les méthodes de conception robuste visent à déterminer les paramètres de conception (dimensions des composants, tension de polarisation, etc.) qui permettent aux circuits d'atteindre les spécifications quelles que soient les dispersions paramétriques. Ces méthodes sont basées sur des méthodes d'analyse de la variabilité et des algorithmes d'optimisation. Cependant, elles présentent en général un coût de calcul élevé ; l'approche traditionnelle consiste donc à effectuer d'abord une conception nominale, puis à appliquer ensuite ces techniques d'amélioration de la robustesse afin de rendre les circuits robustes localement. Cette approche risque alors de ne pas converger vers un dimensionnement globalement robuste. Il est donc nécessaire de mettre en place des méthodes de

conception robuste qui traitent la variabilité dès le début du flot de conception en associant des algorithmes d'optimisation globale avec des techniques de réduction du coût calculatoire.

1.2 Plan du mémoire

Le Chapitre 2 présente les motivations et les limitations de la miniaturisation dans les technologies CMOS. Les sources, le classement, la modélisation, ainsi que l'évolution des phénomènes de variabilité y sont traités en détail.

Dans le Chapitre 3 un état de l'art des méthodes d'analyse de variabilité et de conception robuste des circuits analogiques est proposé.

Le Chapitre 4 présente une nouvelle méthode d'analyse de la variabilité fondée sur les plans d'expériences pour approximer les performances des circuits et le développement limité de Cornish-Fisher pour estimer leurs valeurs extrêmes.

Le Chapitre 5 est consacré à une méthode de modélisation par morceaux qui permet d'approximer les performances des circuits sur des domaines relativement larges.

Dans le Chapitre 6 nous présenterons une méthode de conception robuste basée sur un algorithme d'optimisation globale par intervalles qui fait appel à la modélisation par morceaux présentée dans le Chapitre 5, ainsi qu'à l'approximation de Cornish-Fisher pour estimer les performances pire-cas.

Enfin, le Chapitre 7 constitue la conclusion générale du mémoire et présente les perspectives de recherches qui apparaissent à l'issue de cette thèse.

Chapitre 2 : Les limites des technologies CMOS nanométriques

La formidable croissance qu’a connue l’industrie des semi-conducteurs depuis plus de cinquante ans repose sur un principe fondamental : la miniaturisation. Cependant, alors que les dimensions des composants actuels se rapprochent de celles de la maille cristalline, de plus en plus de limitations à la poursuite de la miniaturisation voient le jour, parmi lesquelles les phénomènes de variabilité. Dans la première partie de ce chapitre, les lois de réduction d’échelle, qui ont régi l’évolution de la microélectronique, sont rappelées. Les limites à la poursuite de la miniaturisation sont ensuite passées en revue. Enfin, les phénomènes de variabilité dans les circuits intégrés, qui sont au cœur de la problématique de cette thèse, sont détaillés.

2.1 La miniaturisation des technologies CMOS

2.1.1 Historique du transistor MOS

A la fin des années 1920, J. E. Lilienfeld fut le premier à poser les bases théoriques des transistors à effet de champ en brevetant plusieurs dispositifs permettant de moduler un courant par une tension. Cependant, il fallut attendre la fin des années 1950 pour que de tels dispositifs voient le jour : le premier transistor MOSFET fut conçu en 1959 par J. Atalla et D. Kahng (laboratoires Bell), quatre ans plus tard, C. T. Sah et F. Wanlass (Fairchild) réalisèrent le premier circuit intégré CMOS (un inverseur comportant deux transistors). Entre-temps, le transistor bipolaire, dont le principe est plus complexe, mais qui était plus facilement réalisable, fut mis au point en 1947 par J. Bardeen, W. Brattain et W. Shockley au sein des laboratoires Bell. Depuis lors, la technologie CMOS s’est imposée devant les autres technologies semi-conducteurs grâce notamment à sa faible consommation statique ; plus de 85 % des circuits intégrés sont aujourd’hui issus d’une filière CMOS.

2.1.2 Loi de Moore

En cinquante ans, les progrès de la microélectronique ont permis de passer d’un circuit intégré à seulement deux transistors à des circuits CMOS logiques de plusieurs centaines de millions de transistors. Cet accroissement exponentiel du nombre de composants par puce a été prédit dès 1965 par G. Moore (cofondateur d’Intel). En 1962, on pouvait intégrer huit transistors sur le même circuit, seize en 1963, trente-deux en 1964 et soixante-quatre en 1965. A partir de ces observations, G. Moore proposa une formule empirique qui prédisait le doublement du nombre de transistors sur une

même surface de circuit intégré tous les ans dans un premier temps [1], puis tous les deux ans [2], pour un coût de fabrication constant (cf. Figure 2.1). Un corollaire important de cette prévision est la baisse du coût d'un transistor du fait de la miniaturisation. Cette célèbre « loi de Moore », qui n'a jamais été mise en défaut jusqu'à maintenant, a servi de fil directeur pour l'industrie des semi-conducteurs en permettant la planification de la recherche technologique par des groupes de travail comme l'ITRS [3].

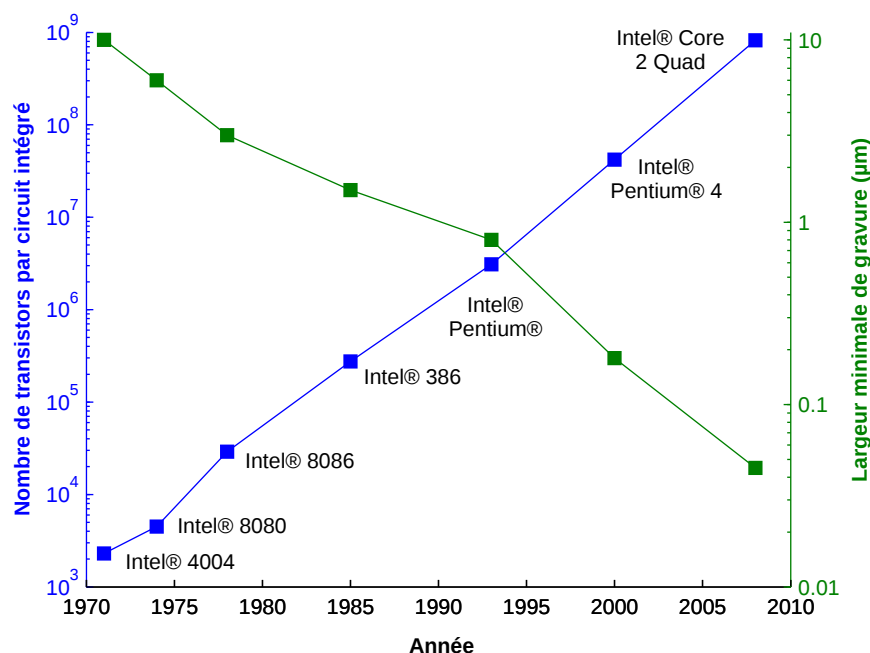


Figure 2.1 : Loi de Moore (source Intel)

2.1.3 Règles de réduction d'échelle de Dennard

La loi de Moore montre l'intérêt de la miniaturisation en termes de coûts de production, cependant ce n'est qu'en 1974 que R. H. Dennard (IBM) exposa une théorie sur la réduction d'échelle des transistors MOS [4] qui révéla tout l'intérêt de la miniaturisation en termes de performances. En effet, d'après la théorie de Dennard, lorsque les tensions et les dimensions d'un transistor MOS sont diminuées d'un facteur α et que les densités de dopage sont augmentées proportionnellement à ce même facteur, alors la configuration des champs électriques à l'intérieur du transistor reste inchangée. Ces relations de réduction d'échelle ainsi que le comportement d'autres paramètres physiques du transistor MOS sont repris dans le Tableau 2.1. En termes de performances, ces règles de réduction d'échelle à champ électrique constant aboutissent à des gains considérables :

- augmentation de la densité d'intégration,
- augmentation de la vitesse de fonctionnement,

- réduction de la puissance dissipée.

Pour l'industrie de la microélectronique, ces améliorations techniques induites par la miniaturisation, présentent trois intérêts forts qui sont à l'origine de la formidable croissance qu'a connue le marché des semi-conducteurs :

- la diminution des dimensions des transistors permet en premier lieu de réduire l'encombrement des systèmes électroniques, facilitant ainsi leur utilisation dans différents secteurs d'activités.
- Ensuite, l'augmentation de la densité d'intégration des circuits, et donc de leur complexité, ainsi que l'augmentation de leur vitesse de fonctionnement contribuent à améliorer leur puissance de calcul, ce qui se traduit par le développement de nouvelles fonctionnalités et donc de nouveaux débouchés.
- Enfin, la réduction d'échelle des transistors, et donc des dimensions des circuits, permet d'augmenter le nombre de puces fabriquées sur une même plaquette de silicium, réduisant ainsi le coût unitaire d'un circuit.

Tableau 2.1 : Règles de réduction d'échelle du transistor MOS proposées par R. H. Dennard

Paramètre physique	Facteur de réduction à champ électrique constant
Dimension physique (longueur du canal, épaisseur de l'oxyde de grille, etc.)	$1/\alpha$
Dopage	α
Tension	$1/\alpha$
Champ électrique	1
Surface	$1/\alpha^2$
Vitesse de fonctionnement	α
Délai de propagation d'une porte	$1/\alpha$
Puissance par circuit	$1/\alpha^2$

Pendant près de quarante ans, les règles de Dennard se sont appliquées avec succès aux générations successives de transistors MOS ; les dimensions minimum des transistors basculant de quelques dizaines de micromètres dans les années 1960 à quelques dizaines de nanomètres au milieu des années 2000. Cependant, au fur et à mesure que les dimensions des transistors se rappro-

chent de celles de la maille cristalline, les règles de réduction d'échelle commencent à présenter certaines limitations ; l'amélioration des performances des circuits et la réduction des coûts de fabrication, qui jusqu'alors ont justifié la course à la miniaturisation, sont de plus en plus délicates à obtenir.

2.2 Les limites de la miniaturisation

La poursuite de la loi de Moore comme fil directeur du développement des circuits CMOS n'est pas sans poser un certain nombre de difficultés. Outre les limites physiques qui surgissent à l'échelle du nanomètre, la complexité croissante des circuits et les investissements colossaux nécessaires à la poursuite de la miniaturisation sont autant d'obstacles à surmonter.

2.2.1 Limites physiques : la barrière de l'atome

L'application des règles de miniaturisation de Dennard aux technologies CMOS a conduit à des dimensions de transistors qui se rapprochent de plus en plus de l'échelle atomique (l'épaisseur d'oxyde de grille d'un transistor MOS submicronique est de l'ordre du nanomètre, soit quelques couches d'atomes). Réaliser des motifs aussi fins pousse les techniques de lithographie classiques dans leurs retranchements. De plus, à ces dimensions-là, des effets physiques secondaires, qui auparavant étaient négligeables, deviennent prépondérants : effets de « canaux courts », courants de fuite dans la grille ou le substrat, saturation de la vitesse des porteurs dans le canal, etc. De ces effets résultent un certain nombre de limitations au niveau des performances des circuits intégrés. Dans la suite de cette section, les principales limitations des technologies CMOS nanométriques et les solutions envisagées sont passées en revue : lithographie, courants de fuite, puissance dissipée, interconnexions, fiabilité. Les dispersions paramétriques sont quant à elles traitées plus amplement dans la partie §2.3.

a) Lithographie

Une étape critique dans la réalisation des circuits intégrés est la photolithographie. Il s'agit d'une technique consistant à projeter sur une résine photosensible l'image d'un masque, correspondant aux motifs du circuit, à l'aide d'un laser. Un paramètre est de grande importance, il s'agit de la résolution qui correspond au plus petit motif (dimension critique « CD ») qui peut être projeté. Afin d'augmenter la résolution, les techniques utilisées jusqu'ici ont consisté à réduire la longueur d'onde du rayonnement utilisé. Ainsi depuis 1980, les longueurs d'onde ont évolué de 436 nm jusqu'à 193 nm, permettant d'atteindre des résolutions inférieures à 45 nm. Depuis le milieu des années 1990, la lithographie est entrée dans un nouveau régime où les dimensions critiques sont

plus petites que les longueurs d'onde utilisées. Ce défi de taille a pu être relevé grâce à l'amélioration des résines photosensibles et des optiques, mais également par la mise en place de techniques d'amélioration de la résolution qui modifient soit la forme de la source lumineuse (« Off-axis illumination ») ou les dessins des masques (« Optical Proximity Correction », « Phase-Shifting Mask ») [5]. Cependant, il devient problématique de diminuer la résolution en continuant de réduire la longueur d'onde du système : la longueur d'onde 157 nm qui devait succéder au 193 nm a été abandonnée face à la difficulté de réaliser les optiques spécifiques. L'alternative qui a été retenue est la lithographie par immersion qui permet d'atteindre une résolution de 32 nm en continuant à utiliser une longueur d'onde de 193 nm pour le laser. Par la suite, la photolithographie à ultraviolet extrême et la lithographie sans masque basée sur un faisceau d'électrons sont des candidats potentiels pour la poursuite de la miniaturisation.

b) Courants de fuite

Avec la réduction d'échelle du transistor MOS, les courants de fuite deviennent de plus en plus significatifs. Il en résulte une dissipation d'énergie alors même que le transistor est à l'état bloqué. Ces courants de fuite ont essentiellement deux origines : le courant de grille et le courant sous le seuil. En effet, la réduction de l'épaisseur de l'oxyde de grille s'accompagne d'une augmentation du champ électrique à travers celui-ci, provoquant le passage de porteurs électriques au travers de l'oxyde par effet tunnel et donc la génération d'un courant parasite de grille. L'utilisation d'isolants dits « high-k » pour l'oxyde de grille, ayant une plus forte permittivité que l'oxyde de silicium et permettant d'obtenir la même capacité de grille avec un oxyde plus épais, constitue une solution pour réduire le courant de grille [6]. En ce qui concerne le courant sous le seuil, il s'agit d'un courant très faible qui apparaît lorsque la tension de grille est inférieure à la tension de seuil (transistor à l'état bloqué), mais qui augmente lorsque la tension de seuil diminue et peut devenir non-négligeable. La réduction des dimensions du transistor allant de pair avec la diminution de la tension de seuil, ce courant de fuite tend à augmenter dans les technologies nanométriques. A cela s'ajoutent les effets dits de « canaux courts » qui participent à l'augmentation du courant sous le seuil : pour des transistors à canaux courts et des potentiels élevés sur le drain, la région de déplétion du drain interagit avec celle de la source pour diminuer sa barrière de potentiel, diminuant ainsi la tension de seuil et aboutissant finalement à l'augmentation du courant sous le seuil. Ces effets de « canaux courts » peuvent être en partie compensés en ajustant le profil de dopage du transistor MOS [7].

c) Puissance dissipée

Paradoxalement, la faible consommation, qui a permis à la technologie CMOS de s'imposer face aux technologies bipolaires, est aujourd'hui sa principale limitation (les processeurs actuels con-

sommant plusieurs centaines de Watts). La puissance totale consommée par un circuit intégré peut se décomposer en une composante active et une composante passive. La composante active correspond à la puissance dissipée lors de la charge et la décharge des capacités du circuit, tandis que la composante passive est due aux courants de fuite à travers chaque transistor comme expliqué précédemment. Ces deux composantes peuvent s'exprimer ainsi :

$$P_{active} = \rho \cdot f \cdot C V_{dd}^2, \quad P_{passive} = I_{off} \cdot V_{dd} \quad (2.1)$$

où ρ est l'activité de commutation, f la fréquence de fonctionnement, C la capacité totale chargée et déchargée en un cycle d'horloge, V_{dd} la tension d'alimentation et I_{off} le courant de fuite global. D'après ces équations, il est clair que l'augmentation de la fréquence de fonctionnement pour accroître la vitesse de calcul et l'augmentation de la surface des puces, donc de la capacité totale, contribuent à augmenter la puissance active dissipée. Par conséquent, le moyen le plus efficace de réduire la puissance active est de diminuer la tension d'alimentation. Cependant réduire la tension d'alimentation impose de réduire également la tension de seuil des transistors MOS, ce qui augmente les courants de fuite (cf. paragraphe précédent) et par là-même la puissance passive dissipée. Pour améliorer la puissance de calcul sans augmenter la fréquence, une solution qui a été retenue consiste à multiplier les cœurs de processeurs ; une architecture parallèle permettant d'augmenter le nombre d'opérations effectuées simultanément en un cycle d'horloge. La puissance dissipée peut, quant à elle, être réduite en adaptant la tension fournie à chaque bloc du circuit intégré en fonction de son activité (techniques de « dynamic voltage scaling » [8, 9]).

d) Interconnexions

Alors que la miniaturisation des dispositifs MOS d'après les règles de Dennard a permis la réduction des délais dans les transistors, elle s'est au contraire accompagnée d'une augmentation des délais dans les interconnexions. Cette limitation peut s'expliquer, d'une part par l'augmentation de la taille des circuits pour augmenter leurs fonctionnalités (ce qui accroît la longueur des interconnexions), et d'autre part par l'augmentation de la résistivité des lignes métalliques avec la réduction de leur épaisseur [6]. De plus, le système d'interconnexions mis en œuvre pour relier des milliards de transistors se complexifiant, les capacités parasites dues au croisement des lignes augmentent également, et avec elles, les pertes capacitatives. Enfin, la taille des circuits augmentant, délivrer un signal sur toute la puce en un cycle d'horloge devient de plus en plus difficile. Par le passé, les solutions mises en place pour améliorer les performances des interconnexions ont consisté à remplacer l'aluminium dans les lignes métalliques par du cuivre dont la résistivité est moindre et à utiliser des diélectriques à faible permittivité pour isoler les différentes couches d'interconnexions, réduisant ainsi les capacités parasites [6]. Malgré cela, les problèmes de consommation et de retard dans les lignes demeurent. Dans l'avenir, les techniques d'intégration 3D, permettant d'empiler les puces les

unes sur les autres, réduisant ainsi la taille des circuits, semblent donc prometteuses [10]. Des interconnexions à base de guides d'onde optique ou de nanotubes de carbone sont également envisagées [11].

e) Fiabilité des transistors MOS

Avec la miniaturisation toujours plus poussée, la fiabilité des composants CMOS devient de plus en plus difficile à assurer. Parmi les sources de dégradation, on peut citer les champs électriques intenses qui, d'une part dégradent lentement la qualité de l'oxyde de grille conduisant à son claquage (« Time Dependent Destructive Breakdown ») et d'autre part altèrent les caractéristiques des transistors tels que le gain de transconductance et la tension de seuil par l'injection de porteurs chauds. Un autre phénomène appelé « Negative Bias Temperature Instability », qui apparaît à des températures élevées et sous certaines conditions de polarisation des transistors PMOS, provoque également un décalage de la tension de seuil. Ces phénomènes de dégradation constituent donc des limitations critiques, notamment pour les circuits analogiques qui requièrent des tensions de seuil stables.

f) Evolution des limites physiques dans les futures technologies CMOS

L'ITRS [3], un groupe de travail composé d'experts internationaux dans le domaine de la microélectronique, met annuellement à jour une feuille de route identifiant les objectifs et les verrous technologiques des futures technologies CMOS. La Figure 2.2, réalisée à partir de l'édition 2009 de cette feuille de route, illustre l'effet de la réduction d'échelle sur différentes caractéristiques technologiques pour les années à venir (2010, 2013 et 2016). Les caractéristiques considérées sont : la longueur physique de la grille des transistors L_G , l'épaisseur d'oxyde de grille T_{OX} , les tensions de seuil V_{TH} et d'alimentation V_{DD} , le délai τ d'un dispositif NMOS, les puissances active P_{active} et passive $P_{passive}$ d'un circuit numérique à haute performance. L'étude des courbes fait clairement apparaître que les règles de réduction d'échelle proposées par R. H. Dennard ne sont plus valables : l'épaisseur de l'oxyde de grille et les tensions ne peuvent pas diminuer aussi rapidement que la longueur de grille afin de limiter les courants de fuite. Malgré cela, la puissance passive dissipée continue d'augmenter.

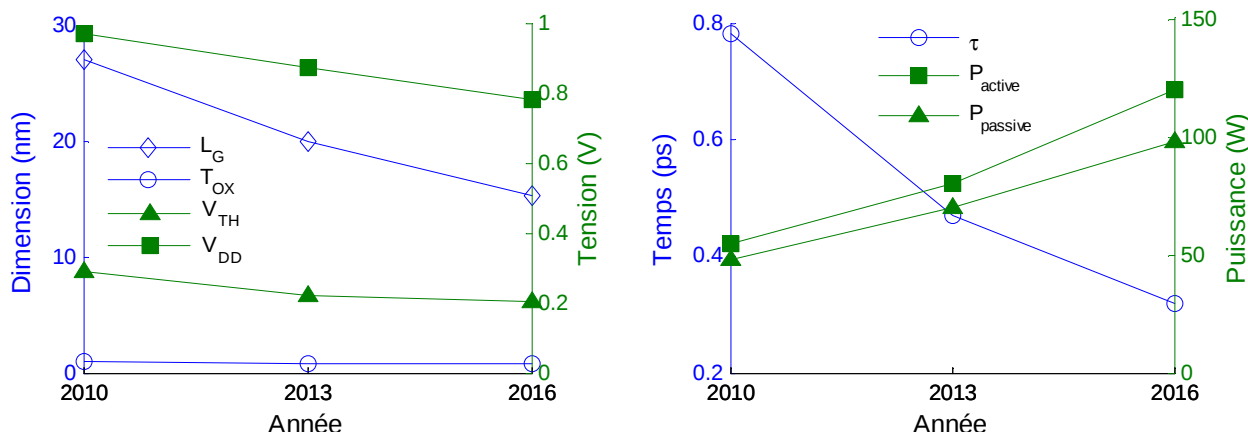


Figure 2.2 : Evolution de différentes caractéristiques des technologies CMOS sur substrat d'après le rapport « Process Integration, Devices & Structures » de l'ITRS 2009

2.2.2 Méthodologies de conception : la maîtrise de la complexité

La miniaturisation, en permettant d'intégrer toujours plus de transistors sur une même puce, a ouvert la voie à des circuits sans cesse plus complexes, favorisant ainsi l'ajout de nouvelles fonctionnalités ; on parle alors de véritables systèmes sur puce (« System on Chip »). Cependant, cet accroissement de la complexité des circuits pose de véritables défis aux méthodes de conception. Les outils de CAO actuels, dont l'objectif est d'améliorer les performances des circuits et réduire les temps de conception pour des coûts toujours moindres, présentent de sérieuses lacunes pour gérer la complexité grandissante des futurs circuits. D'une part, les phénomènes physiques survenant avec la réduction d'échelle nécessitent des modèles de plus en plus complexes, et d'autre part, malgré l'augmentation de la puissance de calcul des ordinateurs, les temps de calcul des outils de conception continuent d'augmenter avec l'intégration d'un nombre toujours plus grand de transistors dans les circuits. Ainsi, les difficultés rencontrées par les méthodes de conception face à l'augmentation de la complexité constituent une limitation majeure à l'accroissement de la densité d'intégration des circuits. L'écart tend même à se creuser entre la complexité des circuits qui augmente de 60 % par an d'après la loi de Moore et la productivité des outils de CAO qui croît seulement de 20 % par an [3]. Cet écart est d'autant plus critique que la consommation de biens électroniques ne cesse de croître et impose des cycles de vie aux produits de plus en plus courts, ce qui nécessite une forte réactivité de la part des industriels pour être les premiers sur le marché (« Time to Market »). Pour accélérer la conception des circuits intégrés, les fabricants de semi-conducteurs ont donc recours à la réutilisation des modules logiciels ou matériels déjà développés et testés (« IP »). A l'avenir, des innovations au niveau logiciel, ainsi qu'au niveau matériel sont nécessaires pour lever les limitations de la miniaturisation sur le flot de conception, mais également pour assu-

rer la robustesse des circuits face aux nouvelles contraintes physiques des technologies CMOS évoquées ci-dessus (dissipation d'énergie, variabilité paramétrique, etc.).

a) Innovation des architectures et des composants

La poursuite de la miniaturisation des technologies CMOS (voie « More Moore » [3]) atteignant ses limites à l'échelle atomique, seule l'intégration de nouveaux matériaux (voie « More than Moore » [3]) permettra d'améliorer les performances des circuits intégrés. Les architectures du futur seront donc des systèmes hétérogènes (« System in Package », intégration 3D), combinant des technologies diverses afin de tirer le meilleur parti de chacune d'elle en fonction de l'application visée : architectures logiques basées sur des transistors à nanotubes de carbone [12], mémoires moléculaires [13], bus d'interconnexions optiques [11], capteurs MEMS/NEMS [14], etc. La prochaine génération de circuits devra également être capable de s'adapter en temps réel aux changements de leur état interne ou de leur environnement, par exemple pour assurer une gestion efficace de leur consommation énergétique. Une voie prometteuse est celle des architectures reconfigurables dynamiquement qui combinent un réseau de blocs élémentaires piloté par du logiciel embarqué [15]. Ces structures présentent une grande flexibilité autant au niveau matériel que logiciel qui leur permet à la fois de s'adapter à plusieurs applications, mais également de minimiser la consommation du circuit ou encore de palier à un dysfonctionnement en remplaçant un bloc défectueux par un autre.

b) Innovation des outils de CAO

Les outils de CAO, déjà confrontés à la complexité des circuits en termes de nombre de transistors, devront également proposer des solutions pour gérer l'hétérogénéité grandissante des architectures et les problèmes d'interface sous-jacents (matériel/logiciel, analogique/numérique). Cette complexité supplémentaire rend la vérification des circuits encore plus problématique, d'où la nécessité de développer des méthodes structurées de conception en vue du test (« Design for Test »). De la même manière, les contraintes physiques des technologies CMOS nanométriques (lithographie, variabilité paramétrique, etc.) doivent être prises en compte au plus tôt dans le flot de conception pour garantir la robustesse des circuits aux variations de la technologie ou de leur environnement et améliorer le rendement de fabrication, on parle alors de « Design for Manufacturing » et de « Design for Yield » [16].

2.2.3 Coût des investissements R&D

Autre obstacle à la réduction d'échelle des transistors : le coût de développement des futures technologies CMOS qui ne cesse d'augmenter. En effet, alors que la mise au point des technologies 90 nm et 45 nm a coûté respectivement 500 millions de dollars et 750 millions de dollars, le coût de

développement de la technologie 32 nm est estimé à 1 milliard de dollars [17]. La valeur d'une usine de production de circuits intégrés est de l'ordre de la dizaine de milliards d'euros, tandis que le coût d'un masque de gravure peut atteindre 1 million de dollars. Il y a principalement trois raisons à cette hausse des coûts de développement. Tout d'abord, la fabrication de dispositifs toujours plus petits nécessite des environnements de production toujours plus propres afin d'éviter que des particules microscopiques contaminent les circuits et les rendent défectueux. Ensuite, la miniaturisation exige des machines de production de plus en plus précises et fiables dont la mise au point et l'entretien s'avèrent par conséquent plus coûteux. Enfin, l'intégration de nouveaux matériaux dans les circuits intégrés pour améliorer leurs performances se traduit par des étapes de production plus nombreuses et plus complexes qui alourdissent le coût de fabrication. Afin de réduire les coûts unitaires de production, l'industrie de la microélectronique cherche à tirer profit au maximum de la fabrication collective en augmentant le diamètre des plaquettes de silicium (300 mm de diamètre, voire 450 mm pour les mémoires) sur lesquelles sont fabriquées les puces. Face à cette augmentation des coûts, différents modèles industriels se développent parmi les fabricants de semi-conducteurs. Ces modèles s'articulent autour des phases de conception et de fabrication des circuits intégrés. La conception comprend les différentes étapes de développement réalisées à partir d'une suite d'outils de CAO. Elle débute avec le cahier des charges du circuit intégré et aboutit à la génération d'un fichier informatique qui décrit le dessin des masques de la puce. Quant à la phase de fabrication, elle a lieu immédiatement après la phase de conception. Elle est effectuée dans une fonderie et consiste à concevoir matériellement la puce à partir du fichier de dessin des masques. Parmi les fabricants de semi-conducteurs, on distingue donc :

- les fondeurs qui se spécialisent dans la phase de fabrication des circuits intégrés (exemple : TSMC),
- les « fabless » qui sous-traitent la fabrication aux fondeurs et concentrent leurs efforts de développement sur la phase de conception (exemple : Qualcomm),
- et les entreprises intégrées qui investissent à la fois dans la conception et la fabrication des circuits intégrés, de plus en plus au sein d'alliances pour partager les coûts de R&D (exemple : alliance IBM).

2.3 Les phénomènes de variabilité

Parmi les limitations physiques des technologies CMOS nanométriques évoquées dans la partie §2.2.1, les phénomènes de variabilité, qui affectent les performances des circuits intégrés, deviennent de plus en plus critiques, notamment ceux issus des procédés de fabrication. En effet, avec la

réduction d'échelle des technologies CMOS, la taille des composants diminue, mais les variations dues au processus de fabrication conservent le même ordre de grandeur. L'impact relatif des variations tend donc à augmenter [18].

2.3.1 Sources des variations

Les circuits intégrés sont affectés par une grande variété de variations. Ces variations se différencient tout d'abord par leur origine. On peut ainsi distinguer [19] :

- **les variations paramétriques** qui résultent du processus de fabrication et affectent les valeurs nominales des paramètres technologiques comme la longueur du canal, l'épaisseur de l'oxyde de grille, la concentration des dopants ou encore les dimensions des interconnexions,
- **les variations des conditions environnementales** lors du fonctionnement du circuit comme la température et la tension d'alimentation,
- **les variations causées par les phénomènes de vieillissement** qui altèrent la fiabilité des composants comme l'injection de porteurs chauds, l'électromigration ou la dégradation NBTI (cf. §2.2.1).

Par la suite, nous nous intéresserons plus particulièrement aux variations paramétriques et à leur impact sur les circuits intégrés.

2.3.2 Classification des variations paramétriques

Il est intéressant de classer les variations des paramètres technologiques suivant leur nature et leur répartition spatiale. On distingue ainsi :

- **les variations systématiques** qui sont causées par des phénomènes physiques clairement identifiés lors du processus de fabrication et dont les tendances de variations en fonction de la position de la puce sur la plaquette de silicium peuvent être modélisées.
- **Les variations aléatoires** qui caractérisent des phénomènes non-déterministes, imprévisibles, et qui peuvent être modélisées par une certaine loi de probabilité.

Concernant leur répartition spatiale, on distingue les variations globales et les variations locales.

- **les variations globales** font référence aux fluctuations d'un paramètre dont la valeur reste constante au niveau d'une même puce mais varie entre deux puces d'une même plaquette de silicium, de plaquettes différentes ou même d'usines de fabrication différentes. Les dispersions globales résultent des variations des procédés de fabrication comme la non-uniformité

des températures de recuit, la variation de focus du système optique, un calibrage de machines légèrement différent entre deux usines de fabrication, etc. Elles présentent en général un caractère systématique, souvent modélisable spatialement sur la plaquette de silicium. Cependant, comme le concepteur ne connaît pas le futur emplacement de sa puce sur la plaquette, les variations globales sont souvent modélisées lors de la conception par des variables aléatoires [19].

- **Les variations locales** correspondent aux fluctuations qui affectent de façon différente les composants d’une même puce et qui contribuent à la perte d’appariement entre des structures identiques (« mismatch »). Avec la miniaturisation des composants, les variations locales ont un impact de plus en plus grand sur le comportement des circuits [20]. Comme pour les variations globales, certaines des variations locales sont causées par des fluctuations lors du processus de fabrication qui engendrent, à la surface de chaque puce, des variations systématiques selon des formes ou des directions précises. Les variations de ce type peuvent être en partie minimisées lors du dessin des masques en utilisant des topologies spécifiques pour les composants, comme les structures « common centroid » pour les paires différentielles d’amplificateurs [21]. Une autre source de variations systématiques est due aux effets de proximité qui dépendent directement du dessin des masques et qui aboutissent à l’altération des dimensions des composants lors des étapes de lithographie ou de planarisation mécano-chimique. Des techniques de correction optique permettent de corriger ces effets de proximité en modifiant la forme du dessin des masques afin de réduire les distorsions dues aux diffractions optiques [6]. Enfin, le point critique pour l’avenir concerne les variations locales causées par des phénomènes purement aléatoires, au premier rang desquels figure la fluctuation statistique du nombre d’atomes de dopage et de leur position dans le canal. En effet, étant donné que le nombre d’atomes de dopage diminue avec la réduction des dimensions, leur impact sur le comportement du transistor augmente ; la tension de seuil des transistors est particulièrement affectée par ce type de fluctuations aléatoires [18, 22]. Parmi les autres dispersions aléatoires, on peut citer les variations de rugosité sur les côtés des structures (« line-edge roughness ») [22, 23] ou les variations de l’épaisseur d’oxyde de grille dont les dimensions avoisinent désormais quelques couches atomiques [22]. Ces variations sont critiques pour les futures technologies CMOS dans la mesure où leur caractère aléatoire ne permet pas de les compenser.

La Figure 2.3 illustre la répartition spatiale des variations ainsi que la part approximative de chaque type de variations dans la variabilité totale. Les données sont issues d’une technologie SOI 65nm d’IBM et publiées dans [24].

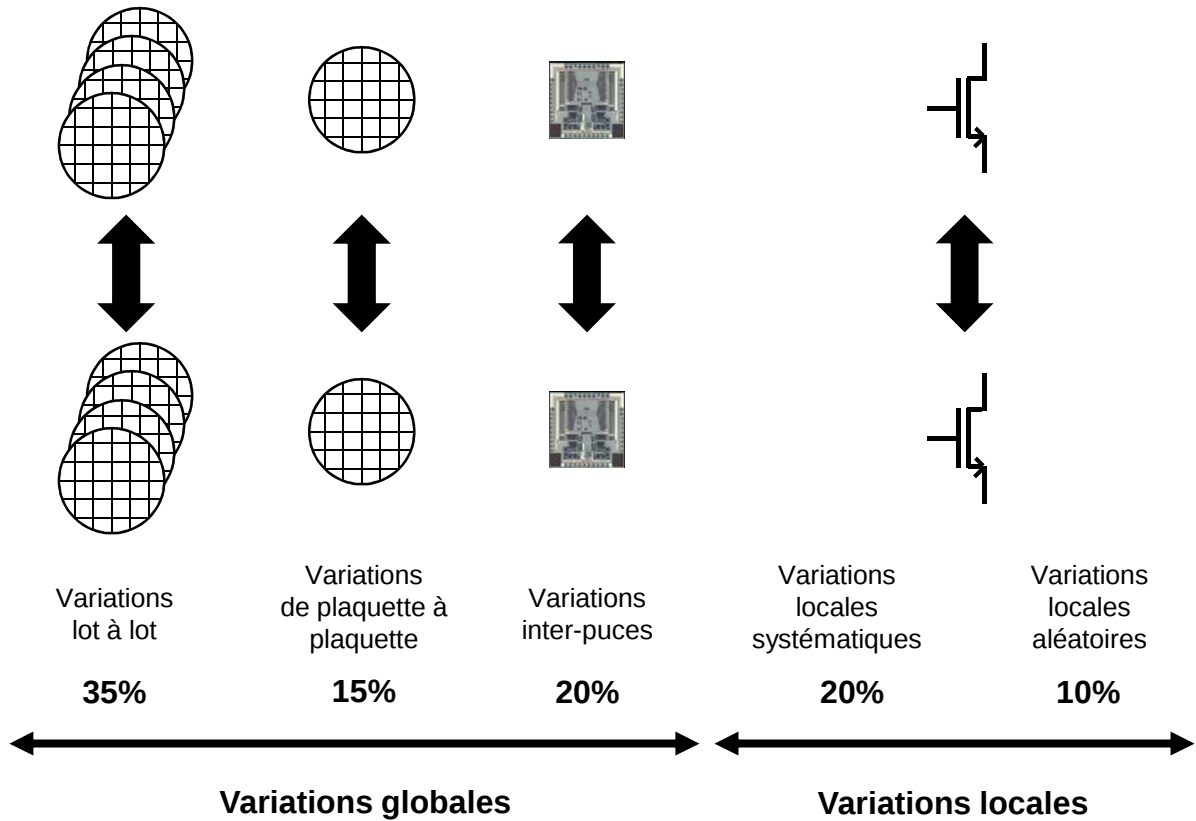


Figure 2.3 : Répartition spatiale des variations paramétriques et pourcentage de la variabilité totale

2.3.3 Modélisation des variations paramétriques

a) Modèle général

Afin d'évaluer l'influence des variations technologiques sur les performances des circuits, le concepteur a besoin de modèles caractérisant les variations des paramètres technologiques en fonction des fluctuations du processus de fabrication. L'approche généralement retenue pour modéliser les variations paramétriques consiste à les décomposer en variations globales et locales. La valeur d'un paramètre technologique s'écrit alors :

$$P = P_{nom} + \Delta P_{globale} + \Delta P_{locale} \quad (2.2)$$

où P_{nom} est la valeur nominale du paramètre, $\Delta P_{globale}$ sa variation globale et ΔP_{locale} sa variation locale. Pour un circuit donné, la valeur de $\Delta P_{globale}$ sera la même pour tous les composants du circuit. En revanche, la valeur de ΔP_{locale} sera propre à chaque composant : le nombre de variations locales à prendre en compte dans un circuit augmente donc proportionnellement avec le nombre de composants.

b) Modélisation des variations globales

Comme nous l'avons vu, les variations globales sont le plus souvent représentées par des variables aléatoires. Ces dernières sont caractérisées par leur distribution et leurs moments. En pratique, une loi normale de moyenne nulle est généralement retenue pour modéliser les variations technologiques. Des mesures sont donc effectuées sur plusieurs puces d'une même plaquette de silicium et appartenant à des plaquettes différentes afin d'en extraire des données statistiques (variance, corrélations, etc.) qui permettront de caractériser les variations globales de chaque paramètre. Le nombre de variations paramétriques à caractériser dépend du modèle des dispositifs retenus (BSIM [25], PSP [26], EKV [27], etc.). Or, certains modèles comportent plusieurs centaines de paramètres souvent corrélés entre eux. Des techniques d'analyse des données sont donc mises en place pour transformer un jeu de n paramètres technologiques en un nouveau jeu de m paramètres avec $m < n$. Ces techniques ont pour objectif la réduction de dimension : soit en réduisant la redondance d'informations comme l'analyse en composantes principales [28, 29] ou l'analyse en composantes indépendantes [30] qui génèrent un nouveau jeu de paramètres décorrélés, soit en appliquant une régression semi-paramétrique comme la régression inverse par tranche (SIR) [31].

c) Modélisation des variations locales

Concernant les variations locales, qui provoquent la perte d'appariement entre deux composants identiques, il a été montré qu'elles dépendaient fortement des dimensions [23, 32]. Le modèle le plus couramment utilisé pour caractériser les variations locales des transistors est le modèle de Pelgrom [32]. Avec ce modèle, la variation locale d'un paramètre technologique est exprimée sous la forme d'une variable aléatoire distribuée selon une loi normale de moyenne nulle et de variance donnée par :

$$\sigma^2(\Delta P_{locale}) = \frac{A_p^2}{WL} + S_p^2 D^2 \quad (2.3)$$

où W et L sont respectivement la largeur et la longueur des transistors appariés, D la distance qui les sépare, enfin A_p et S_p sont des paramètres qui dépendent de la technologie utilisée. Ces deux derniers paramètres sont estimés à partir de mesures réalisées sur des paires de transistors NMOS et PMOS, de dimensions et d'espacements différents. Le premier terme du modèle représente les variations aléatoires, tandis que le deuxième terme permet de tenir compte des variations systématiques. Un corollaire immédiat de ce modèle concerne l'impact des dimensions : plus les dimensions des composants sont grandes, plus la variance des variations aléatoires est faible. C'est la raison pour laquelle les dimensions des dispositifs analogiques d'un circuit sont souvent largement supérieures aux dimensions minimales de la technologie utilisée. Le modèle de Pelgrom est bien adapté pour les transistors à canal long, cependant il ne tient pas compte des effets de « canaux

courts » qui surviennent dans les technologies CMOS nanométriques. D'autres modèles plus précis ont donc été proposés pour tenir compte de ces effets liés à la miniaturisation [33] et assurer la continuité entre les différentes zones de fonctionnement des transistors [34-36].

2.3.4 Evolution de la variabilité avec la miniaturisation

Etant donné le caractère confidentiel des informations techniques sur les performances des futures technologies CMOS, peu de données sont disponibles concernant l'évolution de la variabilité avec la miniaturisation. Néanmoins, l'édition 2009 de la feuille de route de l'ITRS donne un certain nombre d'informations concernant la variabilité de paramètres technologiques clés pour les prochaines années (cf. Figure 2.4). Ces paramètres sont les dimensions minimales CD, la tension d'alimentation V_{DD} , la variabilité de la tension de seuil totale $V_{Th, total}$ et celle due uniquement aux dopants $V_{Th, dopants}$, les délais et la puissance dissipée. D'après les données de l'ITRS, qui ne sont que des recommandations mais qui donnent néanmoins les grandes tendances d'évolution, les innovations technologiques futures permettront de maîtriser la variabilité sur les dimensions minimales et la tension d'alimentation. Cependant, la variabilité sur la tension de seuil continuera à augmenter, les phénomènes de fluctuations aléatoires jouant un rôle de plus en plus prédominant.

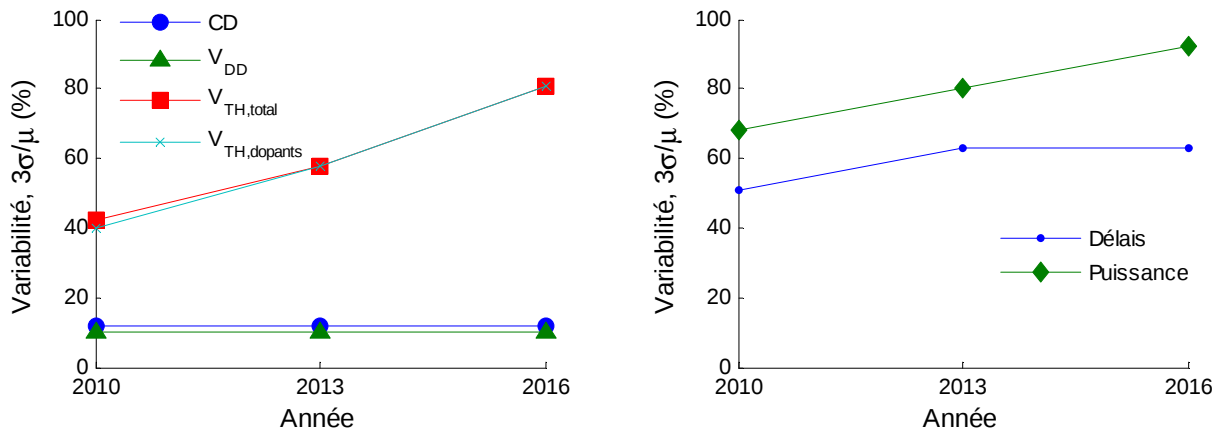


Figure 2.4 : Evolution de la variabilité de différents paramètres dans les futures technologies CMOS d'après l'ITRS. La variabilité d'un paramètre est définie par $3\sigma/\mu$ où μ est sa valeur moyenne et σ son écart-type.

2.4 Conclusion

Depuis les années 1960, la loi de Moore et les règles de miniaturisation de Dennard ont été les principaux vecteurs de la croissance de la microélectronique. Cependant, l'approche de l'échelle atomique pour les dimensions des composants soulève de nouvelles difficultés liées aux limites de la technologie, à la complexité croissante de la conception des circuits et aux coûts des investisse-

ments R&D. Les avantages historiques de la miniaturisation (réduction de l'encombrement, augmentation des fonctionnalités et diminution du coût unitaire des puces) sont plus que jamais contrebalancés par les phénomènes nouveaux qui apparaissent à l'échelle nanométrique (augmentation de la puissance dissipée, baisse de la fiabilité, phénomènes de variabilité). Concernant les dispersions paramétriques, un certain nombre de techniques, tant au niveau de la conception que de la fabrication, permettent de maîtriser les variations systématiques. Cependant, les variations aléatoires, imprévisibles par nature, restent hors de contrôle. Or, l'impact de ces variations tend à augmenter avec la miniaturisation, posant ainsi un véritable défi à la microélectronique.

Chapitre 3 : Etat de l'art des méthodes d'analyse de la variabilité et de conception robuste des circuits analogiques

Nous avons vu dans le chapitre précédent que la miniaturisation des dispositifs CMOS présentait certes de nombreux avantages mais devait également faire face à de nouvelles difficultés, parmi lesquelles les phénomènes de variabilité. A cause de cette variabilité, à l'issue de la fabrication, un certain nombre de circuits ne respectent pas les spécifications définies par le cahier des charges. Cela se traduit par un coût supplémentaire pour les fabricants de semi-conducteurs et donc une baisse de leur compétitivité. Des méthodes de prise en compte de la variabilité, au plus tôt lors de la conception des circuits, sont donc plus que jamais nécessaires pour anticiper les effets de ces variations qui ne font qu'empirer. Dans ce chapitre, deux familles de méthodes dédiées aux circuits analogiques sont passées en revue : les méthodes d'analyse de la variabilité et les méthodes de conception robuste.

3.1 Définitions

3.1.1 Simulation des circuits analogiques

La simulation électrique est une étape incontournable de la conception des circuits analogiques. Son rôle est de prédire les performances des circuits afin de vérifier leurs fonctionnalités. Elle met en œuvre plusieurs éléments qu'il est nécessaire de bien distinguer. Ces éléments sont illustrés sur la Figure 3.1.

Un schéma électrique : il détaille les connexions entre les différents composants (transistors, résistances, condensateurs, sources de tension, etc.) du circuit analogique.

Des modèles mathématiques : ils décrivent le comportement électrique de chaque composant sous forme d'équations. Ces équations font intervenir différents types de paramètres qui sont détaillés ci-dessous. Parmi les modèles de transistors MOS couramment utilisés, on peut citer les modèles BSIM [25], EKV [27] ou PSP [26].

Les paramètres de conception C : ils correspondent aux paramètres des composants que le concepteur peut ajuster pour modifier le comportement du circuit. Il s'agit par exemple des dimensions des transistors, des résistances, des capacités ou des tensions/courants de polarisation.

Les paramètres technologiques T : ils caractérisent la filière CMOS utilisée pour fondre le circuit : épaisseur d'oxyde de grille, tension de polarisation, dopage, etc. Les fluctuations qui surviennent lors du processus de fabrication affectent directement les paramètres technologiques. Ces derniers peuvent donc s'exprimer sous la forme suivante :

$$\mathbf{T} = \mathbf{T}_0 + \Delta\mathbf{T} \quad (3.1)$$

où $\mathbf{T}_0$ représente leur valeur nominale, tandis que $\Delta\mathbf{T}$ modélise leurs variations dues aux fluctuations technologiques. Ces variations peuvent se décomposer en variations locales et globales comme nous l'avons vu dans le Chapitre 2. Dans ce travail de thèse, elles seront modélisées, comme c'est souvent le cas en pratique, par des variables aléatoires normales : $\Delta\mathbf{T} \sim N(0, \Sigma)$ où Σ est la matrice de covariance. A noter que dans le cas des variations locales, la matrice de covariance (et donc la forme de la distribution) dépend des valeurs des paramètres de conception (cf. modèle de Pelgrom).

Les paramètres environnementaux E : ils symbolisent les variations environnementales qui se produisent durant le fonctionnement du circuit comme les fluctuations de la température ou de la tension d'alimentation.

Les performances du circuit Y : comme la consommation, le gain ou la bande passante d'un amplificateur, elles dépendent des paramètres de conception C , des paramètres technologiques T et des paramètres environnementaux E . La relation qui unit les performances aux différents paramètres est généralement complexe ; les performances sont donc évaluées au moyen d'un simulateur électrique.

Un simulateur électrique : il génère les équations qui régissent le comportement électrique du circuit (lois de Kirchhoff), puis résout numériquement ces équations afin d'évaluer les performances. Pour cela, le simulateur a besoin en entrée d'un fichier de description appelé « netlist » qui fusionne toutes les informations concernant le schéma du circuit, les modèles des composants ainsi que les valeurs des différents paramètres. De plus, au moins trois types d'analyses sont généralement proposés dans les simulateurs pour évaluer les différentes sortes de performances : une analyse statique, une analyse transitoire et une analyse « petit signal ». Parmi les simulateurs couramment utilisés, on peut citer Spice [37], Eldo [38] ou Spectre [39].

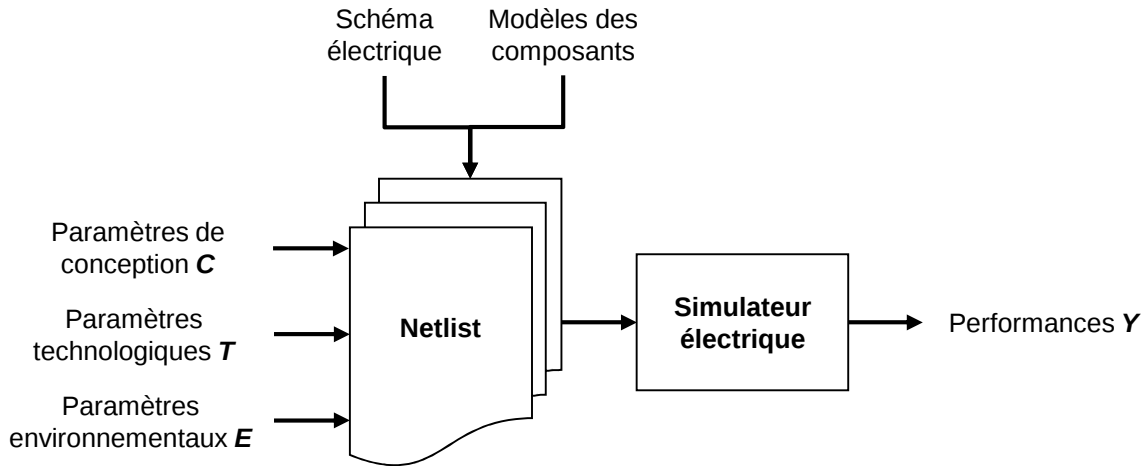


Figure 3.1 : Représentation de la simulation électrique des circuits analogiques

3.1.2 Domaine de conception et régions de tolérance

Les ensembles de valeurs possibles pour les paramètres de conception C et environnementaux E forment respectivement le domaine de conception D et la région de tolérance des paramètres environnementaux RT_E . Ces deux types de paramètres sont des variables réelles définies sur des intervalles fixés par le concepteur. Par exemple, les valeurs possibles pour la longueur de grille d'un transistor pourront être comprises entre une borne inférieure imposée par la taille minimale de gravure et une borne supérieure qui dépendra de la vitesse du circuit. Les ensembles D et RT_E forment donc des hypercubes, également appelés pavés en arithmétiques des intervalles (cf. Annexe C).

De la même manière, on peut définir une région de tolérance notée RT_{AT} pour les variations des paramètres technologiques qui sont modélisées, dans cette thèse, par des variables aléatoires normales de moyenne nulle. Le support d'une variable aléatoire normale de moyenne nulle est infini. Dans la pratique, on considère cependant que l'ensemble des valeurs les plus probables sont comprises dans l'intervalle $[-3\sigma, +3\sigma]$ où σ est l'écart type. En effet, cet intervalle permet d'englober 99.73% des valeurs d'une variable normale centrée, c'est-à-dire la quasi-totalité des valeurs possibles. Comme les ensembles D et RT_E , la région de tolérance RT_{AT} est aussi un pavé (Figure 3.2).

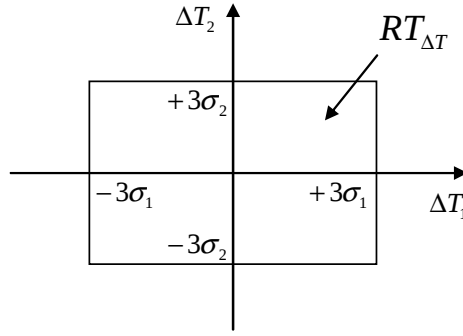


Figure 3.2 : Région tolérance $RT_{\Delta T}$ définie pour deux variations technologiques ΔT_1 et ΔT_2

3.1.3 Régions d'acceptabilité

L'un des objectifs de ce travail de thèse est de déterminer la relation qui unit les performances d'un circuit avec les paramètres de conception, les variations technologiques et environnementales. Soit un circuit comportant m variations de paramètres technologiques et p performances. La relation entre les performances Y et les différents paramètres peut se formuler ainsi :

$$Y = f(C, \Delta T, E) \Leftrightarrow \begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_p \end{pmatrix} = \begin{pmatrix} f_1(C, \Delta T, E) \\ f_2(C, \Delta T, E) \\ \vdots \\ f_p(C, \Delta T, E) \end{pmatrix} \quad (3.2)$$

où $C \in D$, $\Delta T \in RT_{\Delta T}$ et $E \in RT_E$.

Lors de la conception d'un circuit, les fonctionnalités de ce dernier sont exprimées sous la forme de contraintes imposées à ses performances. Ces contraintes sont également appelées les spécifications du circuit. Pour chaque performance, elles sont généralement spécifiées par deux bornes qui définissent un intervalle de valeurs acceptables. La région d'acceptabilité dans l'espace des performances est définie alors par :

$$A_Y = \left\{ Y \in \Re^p \mid \underline{B}_l \leq Y_l \leq \overline{B}_l, \quad l \in \{1, \dots, p\} \right\} \quad (3.3)$$

Il s'agit donc d'un pavé qui est facilement caractérisable.

Etant donné l'impact des variations technologiques sur les performances, il est également intéressant de définir la région d'acceptabilité dans l'espace des variations technologiques. Cette nouvelle région d'acceptabilité est donnée par :

$$\begin{aligned} A_{\Delta T}(C, E) &= \left\{ \Delta T \in RT_{\Delta T} \mid \underline{B}_l \leq Y_l \leq \overline{B}_l, \quad l \in \{1, \dots, p\} \right\} \\ \Leftrightarrow A_{\Delta T}(C, E) &= \left\{ \Delta T \in RT_{\Delta T} \mid \underline{B}_l \leq f_l(C, \Delta T, E) \leq \overline{B}_l, \quad l \in \{1, \dots, p\} \right\} \end{aligned} \quad (3.4)$$

La région d'acceptabilité $A_{\Delta T}(C, E)$ contient toutes les combinaisons possibles des variations technologiques qui, pour des valeurs données des paramètres de conception et environnementaux, conduisent à des performances acceptables. Elle correspond donc à l'image inverse de A_Y par f dans l'espace des paramètres technologiques et dépend des valeurs des paramètres de conception et environnementaux. Les performances f ne sont généralement pas définies de façon explicite et nécessitent le recours à un simulateur électrique, ce qui rend difficile la caractérisation de $A_{\Delta T}(C, E)$. La Figure 3.3 illustre la relation entre ces deux espaces d'acceptabilité pour un circuit comportant deux performances et deux variations technologiques.

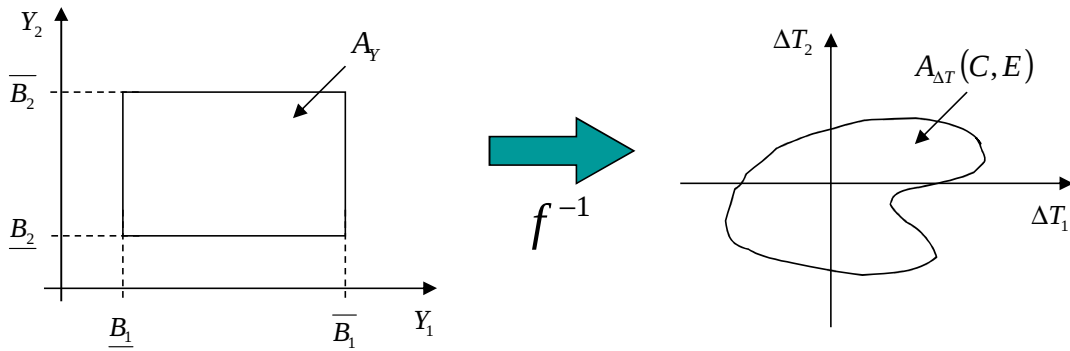


Figure 3.3 : Régions d'acceptabilité A_Y et $A_{\Delta T}(C, E)$ pour deux performances et deux variations technologiques

3.1.4 Rendement paramétrique

L'impact des variations des paramètres technologiques sur le comportement d'un circuit est souvent analysé grâce au rendement de fabrication :

$$\eta_f = \frac{\text{nombre de circuits respectant les spécifications}}{\text{nombre de circuits fabriqués}} \quad (3.5)$$

Cependant, deux types de défaillance peuvent causer le non-respect des spécifications par les circuits : les défaillances catastrophiques et les défaillances paramétriques. Les défaillances catastrophiques résultent d'un dysfonctionnement grave lors de la fabrication (contaminations chimiques, erreur de manipulation, etc.) ; le comportement du circuit est alors totalement aberrant. Quant aux défaillances paramétriques, elles sont dues aux seules fluctuations du processus de fabrication. Par la suite, nous nous intéresserons uniquement aux défaillances paramétriques qui seront caractérisées par un rendement dit paramétrique :

$$\eta = \frac{\text{nombre de circuits respectant les spécifications}}{\text{nombre de circuits fonctionnels}} \quad (3.6)$$

où un circuit est dit fonctionnel s'il n'a pas subi de défaillances catastrophiques. L'histogramme de la Figure 3.4 illustre la notion de circuit fonctionnel, de défaillance paramétrique et catastrophique pour une performance donnée.

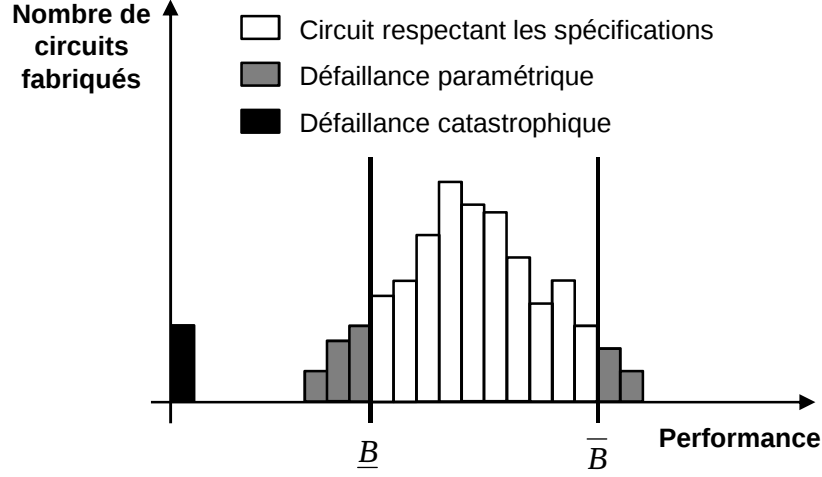


Figure 3.4 : Histogramme de la distribution d'une performance pour plusieurs circuits fabriqués

D'un point de vue mathématique, le rendement paramétrique correspond à la probabilité qu'un circuit respecte toutes les spécifications sur ses performances. Il peut donc s'exprimer à partir de la région d'acceptabilité A_Y :

$$\eta = P(Y \in A_Y) = \int_{A_Y} f_Y(Y_1, \dots, Y_p) dY_1 \dots dY_p \quad (3.7)$$

où f_Y est la densité de probabilité conjointe des performances. Or, comme nous l'avons vu dans la partie §3.1.1, les performances sont évaluées grâce à un simulateur électrique ; leur densité de probabilité n'est donc pas définie de façon explicite. Une autre alternative consiste alors à exprimer le rendement paramétrique à partir de la région d'acceptabilité des variations technologiques $A_{\Delta T}(\mathbf{C}, \mathbf{E})$:

$$\eta = P(\Delta T \in A_{\Delta T}(\mathbf{C}, \mathbf{E})) = \int_{A_{\Delta T}(\mathbf{C}, \mathbf{E})} f_{\Delta T}(\Delta T_1, \dots, \Delta T_m) d\Delta T_1 \dots d\Delta T_m \quad (3.8)$$

où $f_{\Delta T}$ est la densité de probabilité conjointe des variations des paramètres technologiques qui, elle, est connue. Cependant, dans ce cas, la difficulté réside dans la caractérisation de la région d'acceptabilité $A_{\Delta T}(\mathbf{C}, \mathbf{E})$ dont la forme peut s'avérer complexe (cf. Figure 3.3). Les deux expressions (3.7) et (3.8) du rendement paramétrique présentent donc chacune leur inconvénient.

3.2 Etat de l'art - Analyse de la variabilité

3.2.1 Problématique

Comme nous l'avons vu dans le Chapitre 1, la réduction des dimensions dans les technologies CMOS s'accompagne d'une augmentation de la variabilité des paramètres technologiques. Au niveau des circuits, cette variabilité paramétrique va se traduire par une variabilité des performances. On peut alors définir l'analyse de variabilité ainsi :

Définition 1. L'analyse de variabilité a pour objectif d'évaluer l'impact des variations technologiques ΔT sur les performances Y pour un jeu de paramètres de conception donné C_0 .

Nous avons vu que les variations des paramètres technologiques étaient modélisées par des variables aléatoires normales. D'un point de vue mathématique, les performances peuvent donc être également représentées par des variables aléatoires. Toute la difficulté réside dans le fait que leur distribution est généralement inconnue. En effet, étant donné la complexité des modèles de composants utilisés par le simulateur, il est difficile de déterminer la distribution des performances à partir de celles des variations technologiques : les performances extraites avec le simulateur présentant un comportement non-linéaire, leur distribution ne se rapproche d'aucune distribution connue (cf. Figure 3.5).

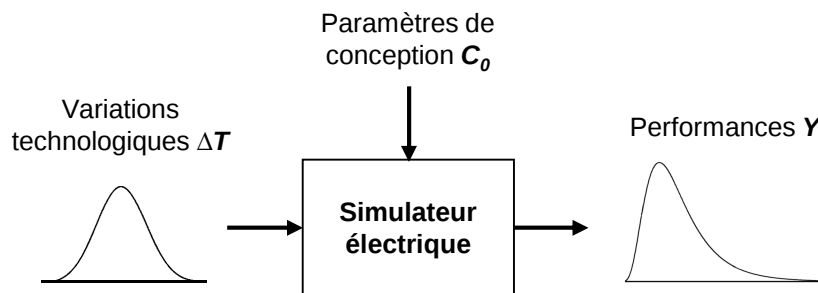


Figure 3.5 : Variabilité des performances causée par les variations paramétriques

Les méthodes d'analyse de la variabilité mettent donc en œuvre des mesures statistiques permettant de caractériser la distribution des performances, comme par exemple ses moments, sa densité de probabilité, ses valeurs extrêmes ou encore son rendement paramétrique. Les principales approches d'analyse de la variabilité sont représentées sur la Figure 3.6. Dans la suite, seule l'influence des variations paramétriques est étudiée, les paramètres environnementaux sont considérés constants.

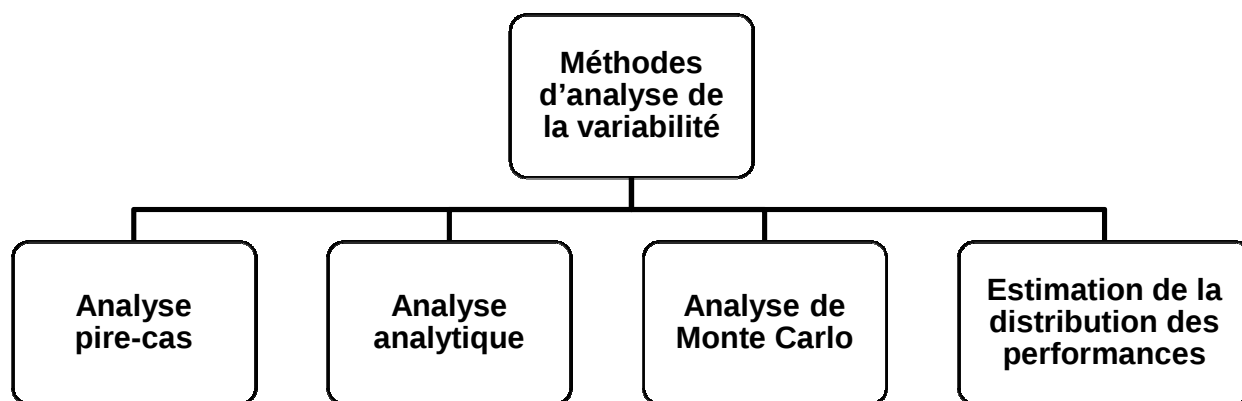


Figure 3.6 : Méthodes d'analyse de la variabilité des circuits analogiques

3.2.2 Analyse pire-cas

L'analyse pire-cas (« worst-case analysis » en anglais) consiste à déterminer les valeurs extrêmes des performances à partir de modèles dits « pire-cas » [40]. Ces modèles pire-cas sont construits en combinant les valeurs extrêmes des paramètres technologiques. Il suffit ensuite de simuler le circuit avec ces modèles pire-cas pour évaluer les valeurs extrêmes des performances. Cette approche présente essentiellement deux avantages :

- il n'est pas nécessaire de connaître la distribution des paramètres technologiques, seule leur région de tolérance RT_{AT} doit être connue pour définir les pire-cas,
- peu de simulations sont effectuées, le temps de calcul est donc faible.

Cependant, l'analyse pire-cas comporte un certain nombre d'inconvénients.

- Les modèles pire-cas, développés par les fabricants de semi-conducteurs et mis à la disposition des concepteurs, sont principalement dédiés aux circuits numériques. En effet, ces derniers ont des topologies semblables, sont conçus avec les plus petites dimensions permises par la technologie et présentent essentiellement deux performances d'intérêt : la vitesse et la consommation, ce qui facilite la mise en œuvre de modèles pire-cas génériques comme les « slow/fast corners ». Ces modèles pire-cas numériques, qui ne tiennent pas compte de la topologie des circuits analogiques, ni de leurs performances, se révèlent de fait mal adaptés à la conception analogique : comment choisir le pire-cas numérique qui sera le plus adapté pour représenter le pire gain ou la pire fréquence de coupure d'un amplificateur ?
- Les corrélations entre les paramètres technologiques ne sont souvent pas prises en compte, ce qui peut conduire à des prédictions très pessimistes (voir Figure 3.7). Des approches amélio-

rées ont néanmoins été proposées afin de construire des modèles pire-cas moins pessimistes [41, 42].

- Les modèles pire-cas ne tiennent pas compte non plus des variations locales dont l'impact est pourtant de plus en plus grand dans les technologies nanométriques.
- Enfin, l'augmentation du nombre de variations technologiques avec la miniaturisation accroît le nombre de combinaisons possibles (2^m où m est le nombre de variations technologiques) et augmente donc le nombre de pire-cas à simuler.

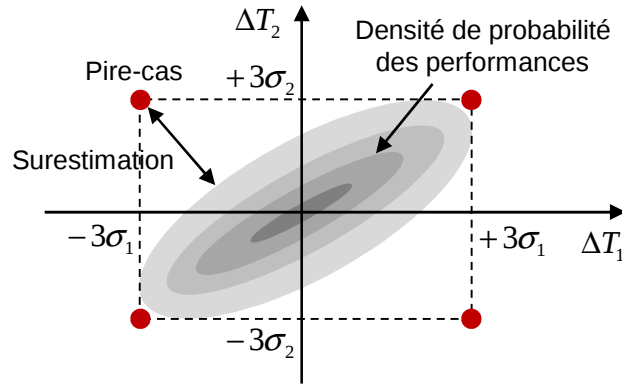


Figure 3.7 : Illustration de la surestimation causée par les modèles pire-cas

3.2.3 Analyse analytique

L'analyse analytique repose sur une analyse de sensibilité et la propagation des moments afin de caractériser la distribution des performances [43-45]. Les performances sont d'abord approximées par un modèle linéaire $\hat{Y}$ en utilisant par exemple un développement de Taylor à l'ordre 1 :

$$\hat{Y} = Y_0 + \sum_{i=1}^m \left[\frac{\partial Y}{\partial \Delta T_i} \right]_0 \Delta T_i \quad (3.9)$$

où Y_0 est la valeur nominale de la performance Y , tandis que les ΔT_i représentent les variations paramétriques qui suivent une loi normale $N(0, \Sigma)$ avec Σ comme matrice de covariance. La distribution de $\hat{Y}$ suit donc également une loi normale dont la moyenne $\mu_{\hat{Y}}$ et la variance $\sigma_{\hat{Y}}^2$ peuvent alors être calculées à partir du modèle linéaire (3.9) :

$$\mu_{\hat{Y}} = Y_0 \quad (3.10)$$

$$\sigma_{\hat{Y}}^2 = \sum_{i=1}^m \left[\frac{\partial Y}{\partial \Delta T_i} \right]_0^2 \sigma_{\Delta T_i}^2 + 2 \sum_{i=1}^{m-1} \sum_{q=i+1}^m \left[\frac{\partial Y}{\partial \Delta T_i} \right]_0 \left[\frac{\partial Y}{\partial \Delta T_q} \right]_0 \text{cov}(\Delta T_i, \Delta T_q) \quad (3.11)$$

L'avantage majeur de cette approche est son faible coût de calcul pour caractériser la distribution des performances. Cependant, dès que les performances tendent à être non-linéaires et les variations de moins en moins faibles, l'approximation linéaire n'est plus adaptée et conduit à des erreurs d'estimation. Néanmoins, cette approche peut fournir une première indication sur la variabilité des performances.

3.2.4 Analyse de Monte Carlo

La plupart des méthodes fondées sur l'analyse de Monte Carlo furent développées pour les circuits discrets au début des années 1980 ; il s'agit donc de l'approche classique d'analyse de la variabilité. Elle est basée sur l'échantillonnage de la région de tolérance des variations technologiques RT_{AT} en respectant leur distribution grâce à un générateur de variables aléatoires. Elle permet d'évaluer deux mesures statistiques importantes qui définissent les variations des performances : l'histogramme de leur distribution et le rendement paramétrique.

a) Histogramme des performances

L'histogramme est un estimateur relativement simple pour approximer la forme de la densité de probabilité d'une variable aléatoire. Dans notre cas, il est construit à partir des valeurs de performances obtenues en chaque point échantillonné de la région de tolérance RT_{AT} : plus le nombre d'échantillons est grand, plus l'histogramme tend vers la forme de la densité de probabilité des performances. Les performances peuvent être évaluées avec un simulateur électrique (Spice, Eldo, etc.) ou à partir d'un méta-modèle.

- L'analyse de Monte Carlo avec un simulateur est l'approche la plus précise et sert souvent de méthode de référence. Cependant, pour obtenir une précision suffisante, plusieurs milliers d'échantillons sont nécessaires, ce qui requiert autant de simulations électriques ; le temps de calcul devient donc rapidement prohibitif.
- L'analyse de Monte Carlo avec un méta-modèle permet de réduire le coût calculatoire et constitue une alternative plus efficace. L'idée est d'approximer les performances par des méta-modèles (moins précis que le simulateur, mais plus rapides à simuler), puis d'utiliser ceux-ci à la place du simulateur pour évaluer les performances. Plusieurs types de méta-modèles sont envisageables : polynômes [28, 30], modèles symboliques [46], krigeage [47, 48], réseaux de neurones [49, 50]. La construction de méta-modèles polynomiaux à partir des plans d'expériences est expliquée dans le Chapitre 4.

b) Rendement paramétrique

Le rendement paramétrique (3.7) est défini par une intégrale. Une solution envisageable pour calculer cette intégrale serait d'avoir recours à des méthodes de quadrature numérique. Cependant, le coût calculatoire de ces méthodes augmente de façon exponentielle avec le nombre de variations technologiques. L'analyse de Monte Carlo est, au contraire, peu sensible au nombre de variables en jeu [19], c'est pourquoi elle est traditionnellement utilisée pour calculer le rendement paramétrique des circuits.

Fonctions indicatrices

L'analyse de Monte Carlo permet d'estimer le rendement paramétrique au moyen des fonctions indicatrices suivantes :

$$I_Y(\mathbf{Y}) = I_Y(Y_1, \dots, Y_p) = \begin{cases} 1 & \text{si } \mathbf{Y} \in A_Y \\ 0 & \text{si } \mathbf{Y} \notin A_Y \end{cases} \quad (3.12)$$

$$I_{\Delta T}(\Delta \mathbf{T}) = I_{\Delta T}(\Delta T_1, \dots, \Delta T_m) = \begin{cases} 1 & \text{si } \Delta \mathbf{T} \in A_{\Delta T}(\mathbf{C}) \\ 0 & \text{si } \Delta \mathbf{T} \notin A_{\Delta T}(\mathbf{C}) \end{cases} \quad (3.13)$$

Ces fonctions décrivent l'appartenance d'un circuit aux régions d'acceptabilité et traduisent ainsi le fait qu'un circuit respecte ou pas les spécifications. La première s'appuie sur la région d'acceptabilité dans l'espace des performances A_Y et la seconde sur la région d'acceptabilité dans l'espace des variations technologiques $A_{\Delta T}(\mathbf{C})$. Avec ces fonctions indicatrices, les expressions du rendement paramétrique (3.7) et (3.8) s'écrivent alors :

$$\begin{aligned} \eta &= \int_{A_Y} f_Y(Y_1, \dots, Y_p) dY_1 \dots dY_p \\ \eta &= \int_{-\infty}^{+\infty} I_Y(Y_1, \dots, Y_p) f_Y(Y_1, \dots, Y_p) dY_1 \dots dY_p \end{aligned} \quad (3.14)$$

$$\eta = E(I_Y(\mathbf{Y}))$$

$$\begin{aligned} \eta &= \int_{A_T(\mathbf{C}, E)} f_{\Delta T}(\Delta T_1, \dots, \Delta T_m) d\Delta T_1 \dots d\Delta T_m \\ \eta &= \int_{-\infty}^{+\infty} I_{\Delta T}(\Delta T_1, \dots, \Delta T_m) f_{\Delta T}(\Delta T_1, \dots, \Delta T_m) d\Delta T_1 \dots d\Delta T_m \\ \eta &= E(I_{\Delta T}(\Delta \mathbf{T})) \end{aligned} \quad (3.15)$$

Les fonctions indicatrices permettent donc d'écrire l'intégrale du rendement sous la forme d'une valeur moyenne qui peut ensuite être estimée à partir d'une analyse de Monte Carlo. En effet, en disposant de N échantillons aléatoires des performances $(\mathbf{Y}^1, \dots, \mathbf{Y}^N)$ ou des variations technolo-

giques $(\mathbf{AT}^1, \dots, \mathbf{AT}^N)$, les expressions (3.14) et (3.15) peuvent être approchées en calculant la moyenne empirique sur ces N échantillons :

$$\eta = E(I_Y(\mathbf{Y})) = \frac{1}{N} \sum_{i=1}^N I_Y(\mathbf{Y}^i) = \frac{n_{A_Y}}{N} \quad (3.16)$$

$$\eta = E(I_{AT}(\mathbf{AT})) = \frac{1}{N} \sum_{i=1}^N I_{AT}(\mathbf{AT}^i) = \frac{n_{A_{AT}(\mathbf{C})}}{N} \quad (3.17)$$

où n_{A_Y} est le nombre d'échantillons des performances qui appartiennent à A_Y et $n_{A_{AT}(\mathbf{C})}$ le nombre d'échantillons des variations technologiques qui appartiennent à $A_{AT}(\mathbf{C})$. On retrouve alors l'expression (3.6) du rendement paramétrique. La variance de l'estimateur de Monte Carlo est proportionnelle à $N^{-1/2}$ [19] ; le nombre d'échantillons N doit donc être suffisamment élevé pour garantir une précision correcte.

Ainsi, deux approches sont possibles pour évaluer le rendement paramétrique avec une analyse de Monte Carlo : raisonner dans l'espace des performances A_Y ou dans la région d'acceptabilité de l'espace des variations technologiques $A_{AT}(\mathbf{C})$.

Calcul du rendement par évaluation des performances

La première approche consiste à générer N échantillons des variations technologiques en respectant leur distribution, puis à évaluer les performances pour chacun d'eux, de façon à obtenir N échantillons des performances $(\mathbf{Y}^1, \dots, \mathbf{Y}^N)$. L'appartenance des performances à la région d'acceptabilité A_Y est ensuite testée afin de calculer le rendement grâce à l'expression (3.16). La région A_Y étant définie par un pavé (cf. Figure 3.3), il est aisé de tester l'appartenance des performances à A_Y . Avec cette approche, l'effort de calcul se concentre donc dans l'évaluation des performances. Là encore, le recours à des méta-modèles à la place d'un simulateur permet de réduire les temps de calcul.

Calcul du rendement par approximation de la région d'acceptabilité

La seconde approche consiste à construire une approximation $\hat{A}_{AT}(\mathbf{C})$ de la région d'acceptabilité $A_{AT}(\mathbf{C})$, puis à générer N échantillons des variations technologiques en respectant leur distribution et enfin à tester leur appartenance à $\hat{A}_{AT}(\mathbf{C})$ pour évaluer le rendement avec l'expression (3.17). Ici, toute la difficulté réside dans l'approximation de $A_{AT}(\mathbf{C})$ dont la forme peut-être complexe (cf. Figure 3.3). Des méthodes d'approximation polyédrique [51, 52] et ellipsoïdale [53, 54] faisant appel à des algorithmes d'optimisation ont donc été développées (cf. Figure

3.8). Cependant, ces méthodes présentent plusieurs inconvénients : tout d'abord elles se révèlent mal adaptées lorsque la région d'acceptabilité n'est pas convexe, de plus elles ne donnent pas d'informations sur la précision de l'approximation obtenue.

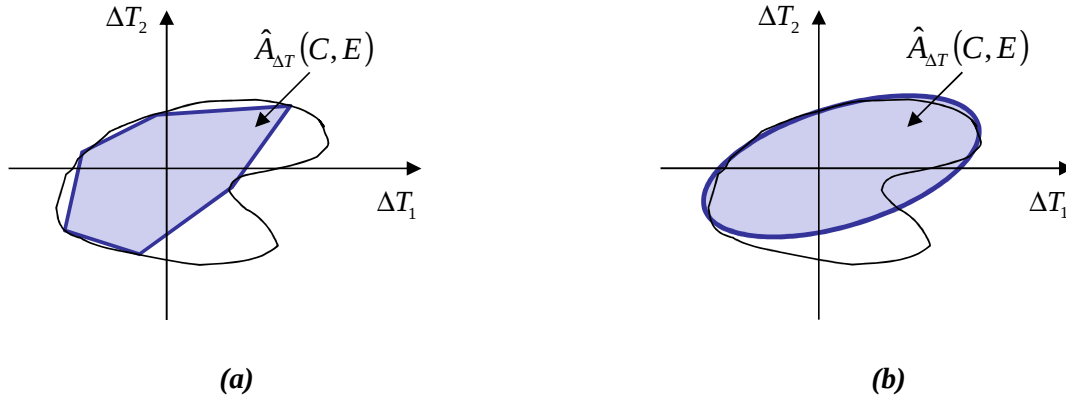


Figure 3.8 : Approximations polyédrique (a) et ellipsoïdale (b) de la région d'acceptabilité

Signalons enfin pour être complet, l'existence d'autres méthodes heuristiques d'estimation du rendement comme l'exploration orthogonale [55] et radiale [56], initialement mises au point pour les circuits discrets, ou encore la méthode d'intégrale de surface [57].

c) Bilan

Les principaux avantages de l'analyse de Monte Carlo sont :

- la simplicité de mise en œuvre (aucune restriction sur la forme des performances ou la façon de les évaluer),
- la distribution des variations technologiques ne doit pas nécessairement être normale,
- la précision obtenue qui ne dépend pas du nombre de paramètres en jeu.

Cependant, l'inconvénient majeur reste le nombre élevé d'échantillons qui est nécessaire pour garantir la précision de l'approche. Pour palier cela, des méthodes d'échantillonnages appropriées peuvent être mises en place pour réduire le nombre d'échantillons (échantillonnage d'importance, échantillonnage stratifié).

3.2.5 Estimation de la distribution des performances

Les méthodes d'estimation d'une distribution n'ont été appliquées que récemment à l'analyse de variabilité des circuits analogiques (années 2000). D'une manière générale, leur but est d'estimer la

distribution des performances à partir d'un nombre restreint d'évaluations, c'est-à-dire pour un coût de calcul nettement inférieur à celui d'une analyse de Monte Carlo classique.

a) Identification par les moments

L'approche APEX proposée par X. Li dans [59] permet d'approximer la densité de probabilité d'une performance. Le point fondamental de cette approche consiste à modéliser la densité de probabilité comme étant la réponse impulsionnelle d'un système linéaire et invariant dans le temps. La fonction de transfert d'un tel système à l'ordre M dans le domaine de Laplace et sa réponse impulsionnelle s'écrivent respectivement ainsi :

$$H(p) = \sum_{i=1}^M \frac{a_i}{p - b_i} \quad (3.18)$$

$$h(t) = \begin{cases} \sum_{i=1}^M a_i e^{b_i t} & \text{si } t \geq 0 \\ 0 & \text{si } t < 0 \end{cases} \quad (3.19)$$

La méthode APEX se déroule en trois étapes :

- la performance est d'abord approximée par un modèle quadratique,
- les moments statistiques de la performance sont ensuite calculés à partir des coefficients du modèle quadratique,
- la résolution du système d'équations non-linéaires obtenu en identifiant les moments de H avec les valeurs obtenues précédemment permet finalement de calculer les coefficients a_i et b_i de la réponse impulsionnelle $h(t)$, et donc d'approximer la densité de probabilité par $h(t)$.

Grâce à l'approximation de la densité de probabilité, les valeurs pire-cas de la performance (quantiles extrêmes de sa distribution) sont calculées. D'après les résultats obtenus sur des circuits analogiques [59], la méthode APEX se révèle autant précise que l'analyse de Monte Carlo pour un coût de calcul divisé par 10. Cependant cette méthode ne reste efficace que si les variations technologiques sont modélisées par des variables aléatoires normales.

b) Approximation des queues de distribution

Une autre approche particulièrement intéressante pour estimer les valeurs pire-cas des performances consiste à approximer non pas la totalité de la distribution, mais uniquement les queues de distribution ; cette approche est basée sur la théorie des valeurs extrêmes dont le but est d'estimer la probabilité d'événements rares [60, 61]. Elle repose sur l'extrapolation de la queue d'une distribution à partir des valeurs extrêmes des données disponibles. De nombreux domaines s'intéressent à

ce type d'analyse afin d'anticiper des événements d'autant plus catastrophiques qu'ils sont exceptionnels : hydrologie (crues), climatologie (réchauffement climatique), finance (crises boursières), etc. La théorie des valeurs extrêmes a également été appliquée récemment à l'analyse de variabilité des circuits, en particulier pour estimer la probabilité de défaillance des mémoires SRAM [62-64]. La probabilité de défaillance est illustrée sur la Figure 3.9. A la différence des phénomènes naturels, où le nombre d'observations disponibles est imposé par l'environnement et n'est pas maîtrisable, les simulateurs électriques permettent d'évaluer les performances d'un circuit pour n'importe quelle combinaison des variations technologiques. Il est donc intéressant d'utiliser cette souplesse afin de choisir les combinaisons de variations technologiques qui vont conduire aux valeurs extrêmes des performances et à partir desquelles la queue de distribution va pouvoir être estimée. L'approche proposée dans [64] consiste ainsi à :

- générer des échantillons aléatoires des variations technologiques selon leur distribution (Monte Carlo),
- puis identifier ceux qui appartiennent à la queue de distribution grâce à une technique de classification de type SVM (Machine à vecteurs de support) et les simuler,
- enfin approximer la queue de la distribution à partir des échantillons simulés en appliquant la théorie des valeurs extrêmes et ainsi calculer la probabilité de défaillance.

L'approximation des queues de performances présente de nombreux avantages : pas de contraintes sur la forme des distributions des variations technologiques, une très bonne estimation de la probabilité de défaillance calculée pour des quantiles extrêmes, voire même très éloignés ($\pm 6\sigma$), enfin un coût de calcul relativement faible (une centaine de simulations dans [63]) par rapport à une analyse de Monte Carlo classique.

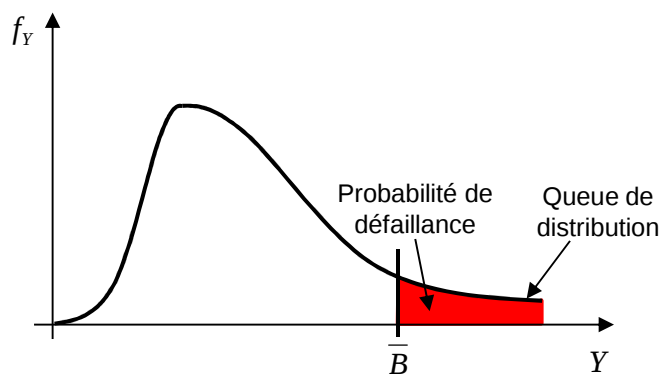


Figure 3.9 : Illustration de la probabilité de défaillance étant donné une spécification $\bar{B}$ sur une performance Y

3.2.6 Bilan

Les différentes méthodes d'analyse de la variabilité ont été comparées dans le Tableau 3.1. Les principaux critères de comparaison sont l'indicateur statistique de la variabilité, la précision, le coût calculatoire et les éventuelles conditions sur la distribution des variations technologiques. Il ressort de cette étude que les méthodes historiques (analyse pire-cas, analyse de Monte Carlo) montrent de sérieuses limites en termes de coût de calcul ou de précision avec l'avènement des technologies nanométriques. Les méthodes les plus récentes s'attachent donc à maintenir une grande précision en réduisant autant que possible les temps de calcul. L'approche qui apparaît comme la plus efficace est l'approximation des queues de distributions : elle associe précision et efficacité en n'imposant aucune contrainte sur le type de distribution des variations technologiques.

Tableau 3.1 : Méthodes d'analyse de la variabilité

Méthodes	Mesure de variabilité	Avantages	Inconvénients
Pire-cas	- Valeurs extrêmes des performances	- pas de contraintes sur la forme de la distribution des variations technologiques - coût de calcul faible - disponible dans la plupart des outils de CAO	- pas adaptée aux circuits analogiques - estimation pessimiste - ne tient pas compte des corrélations et des variations locales
Analytique (modèle linéaire)	- Distribution des performances - Rendement paramétrique	- coût de calcul très faible	- pas adaptée aux performances fortement non-linéaires et aux larges variations
Monte Carlo	- Distribution des performances - Rendement paramétrique	- très bonne précision indépendamment du nombre de paramètres - pas de contraintes sur la forme de la distribution des variations technologiques - disponible dans la plupart des outils de CAO	- coût de calcul considérable (améliorations possibles avec un échantillonnage adapté et l'approximation des performances par des méta-modèles)
Identification par les moments (APEX)	- Distribution des performances - Rendement paramétrique	- très bonne précision - coût de calcul faible	- la distribution des variations technologiques doit être normale
Approximation des queues de distributions	- Probabilité de défaillance	- très bonne précision - coût de calcul faible - pas de contraintes sur la distribution des variations technologiques	

3.3 Etat de l'art - Méthodes de conception robuste

3.3.1 Problématique

A cause de la variation des performances, due à la variabilité des paramètres technologiques, les spécifications des circuits risquent de ne pas être respectées, entraînant ainsi une baisse du rendement paramétrique. Il est donc important de considérer la variabilité des performances le plus tôt possible dans le cycle de conception afin de fabriquer des circuits qui soient robustes aux variations : c'est l'objectif de la conception robuste.

Définition 2. La conception robuste a pour objectif d'ajuster les paramètres de conception C de telle sorte que les spécifications sur les performances Y soient respectées quelles que soient les fluctuations paramétriques ΔT appartenant à la région de tolérance $RT_{\Delta T}$.

Dans le flot de conception analogique, la robustesse est généralement prise en compte après une première étape de conception nominale. Cette étape consiste à dimensionner le circuit pour qu'il respecte les spécifications sans tenir compte des variations paramétriques ; elle peut être manuelle ou automatisée. Depuis le début des années 1980, de nombreux outils de dimensionnement automatisé ont vu le jour afin de réduire le coût de la conception nominale et d'améliorer la qualité des circuits analogiques [65]. Comme la conception nominale, la conception robuste est également fondée sur des méthodes d'automatisation qui consistent à optimiser une mesure caractéristique de la variabilité : les méthodes de conception robustes sont donc souvent basées sur les méthodes d'analyse de la variabilité présentées dans la partie précédente. La mesure statistique la plus souvent optimisée dans les méthodes de conception robuste est le rendement paramétrique. Cependant, d'autres mesures existent comme les performances pire-cas ou les indices de capabilité. La Figure 3.10 propose un classement des différentes méthodes de conception robuste.

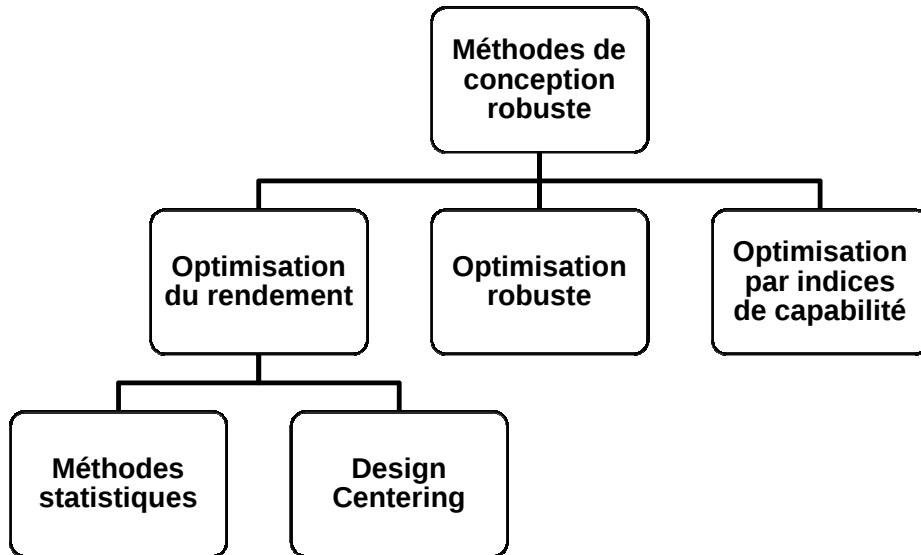


Figure 3.10 : Méthodes de conception robuste des circuits analogiques

3.3.2 Circuits discrets

Historiquement, les premières méthodes de conception robuste ont été développées pour des circuits discrets. Pour ces circuits, la notion de paramètres technologiques n'existe pas ; la seule variabilité prise en compte est celle des paramètres de conception :

$$C = C_0 + \Delta C \quad (3.20)$$

où C_0 est la valeur nominale d'un paramètre de conception (résistance par exemple) et ΔC sa variation modélisée par une variable aléatoire (tolérance). Pour les circuits discrets, on peut donc définir une région d'acceptabilité A_C directement dans l'espace des paramètres de conception (voir Figure 3.11).

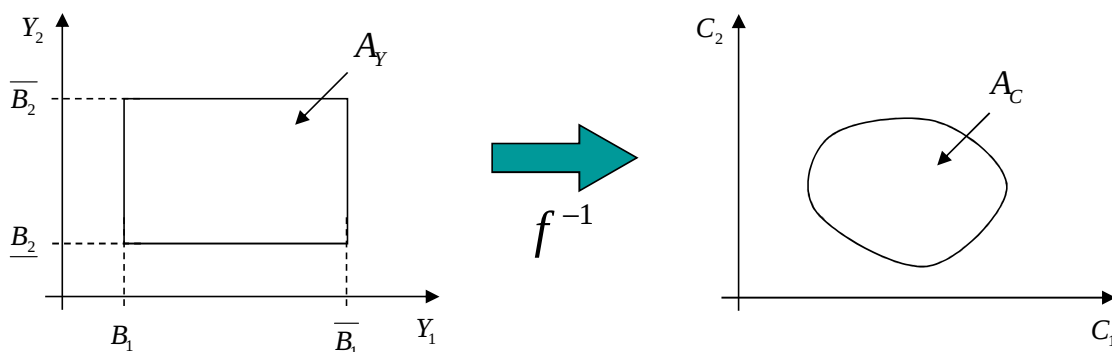


Figure 3.11 : Régions d'acceptabilité A_Y et A_C pour deux performances et deux paramètres de conception dans le cas des circuits discrets

3.3.3 Optimisation du rendement

Le problème de la conception robuste est souvent traité en optimisant le rendement paramétrique. Les méthodes d'optimisation du rendement ont pour but de déterminer les paramètres de conception C qui maximisent le rendement paramétrique. Ces méthodes s'articulent autour de deux étapes fondamentales : l'évaluation du rendement et son optimisation. On peut distinguer deux grandes classes de méthodes d'optimisation du rendement : les méthodes statistiques et les méthodes déterministes.

a) Méthodes statistiques

Dans les méthodes statistiques, le rendement paramétrique est la fonction de coût du problème d'optimisation, c'est pourquoi ces méthodes sont également appelées méthodes directes dans la littérature. Elles ont en commun l'évaluation du rendement qui est effectuée par un échantillonnage de Monte Carlo. Les raisons qui motivent le choix de l'analyse de Monte Carlo ont été expliquées dans la partie §3.2.4 : il s'agit de la possibilité de traiter indifféremment n'importe quel type de distribution et de la relative indépendance de la complexité vis-à-vis du nombre de paramètres en jeu. Les méthodes statistiques se distinguent entre elles suivant la procédure qui est mise en place pour obtenir le meilleur rendement possible.

Algorithmes d'optimisation

L'approche la plus simple consiste à avoir recours à un algorithme déterministe d'optimisation non-linéaire pour maximiser le rendement. Bien évidemment, effectuer une analyse de Monte Carlo avec un simulateur à chaque itération de l'algorithme alourdit considérablement le coût calculatoire et rend cette approche impraticable. Les méthodes traditionnelles pour réduire le coût d'une analyse de Monte Carlo peuvent être appliquées comme l'échantillonnage d'importance [66] ou l'approximation des performances par des modèles polynomiaux [67, 68] ou des réseaux de neurones [49]. Une autre solution pour réduire les temps de calcul est d'évaluer le gradient du rendement [57, 69] afin d'appliquer des algorithmes d'optimisation plus efficaces qui convergent rapidement. Néanmoins, l'estimation du rendement et de son gradient par une analyse de Monte Carlo est entachée de bruit numérique, ce qui ralentit la convergence de l'algorithme et peut même le faire diverger. Pour remédier à cela, des méthodes stochastiques permettent de tenir compte de l'incertitude sur l'estimation du rendement [70, 71]. Par ailleurs, excepté l'approche présentée dans [49] qui fait appel à un algorithme génétique, les algorithmes utilisés sont généralement des algorithmes d'optimisation locaux qui conduisent par définition à un rendement maximum local.

Algorithmes heuristiques

D'autres méthodes d'optimisation du rendement ne sont pas basées sur des algorithmes d'optimisation, mais ont une approche heuristique du problème. Ces méthodes ont initialement été mises au point pour des circuits discrets (cf. §3.3.2). Parmi celles-ci, on peut citer la méthode proposée par [58, 72] qui consiste à échantillonner la région de tolérance des paramètres de conception selon la distribution de leurs variations, puis à calculer les centres de gravité des points acceptés (qui appartiennent à la région d'acceptabilité) et celui des points rejetés (qui n'appartiennent pas à la région d'acceptabilité). La proportion de points acceptés par rapport au nombre total de points générés fournit une approximation du rendement. Un nouveau jeu de valeurs pour les paramètres de conception est alors choisi dans la direction définie par les deux centres de gravité afin de déplacer la région de tolérance dans la région d'acceptabilité et ainsi augmenter le rendement. Le principe de l'algorithme des centres de gravité est illustré sur la Figure 3.12. Une autre méthode heuristique introduite par [56] consiste également à déplacer la région de tolérance dans la région d'acceptabilité, mais est basée sur l'estimation du rendement par l'exploration radiale. Le dimensionnement qui sert de point de départ à ces méthodes ne doit pas présenter un rendement nul, sinon l'algorithme risque de ne pas converger ; ces méthodes heuristiques sont donc plutôt considérées comme des méthodes d'amélioration du rendement que d'optimisation.

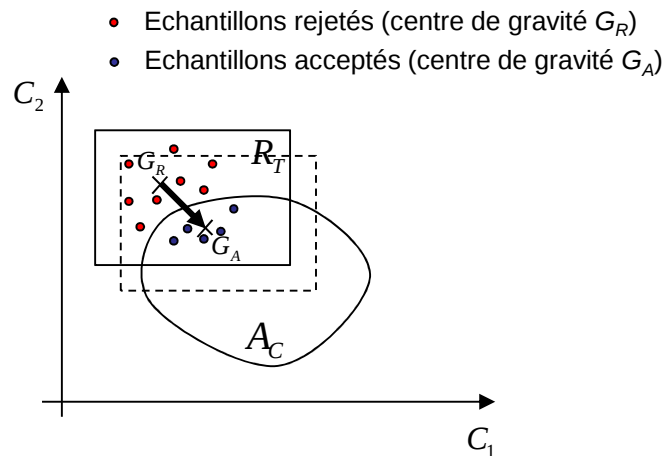


Figure 3.12 : Déplacement de la région de tolérance R_T avec la méthode des centres de gravité

b) Méthodes déterministes dites de « design centering »

Comme nous venons de le voir, l'optimisation directe du rendement paramétrique s'avère très lourde en termes de temps de calcul. Des méthodes déterministes ont donc été développées afin d'optimiser le rendement de façon indirecte en raisonnant sur le positionnement des distributions par rapport aux régions d'acceptabilité. En effet, le rendement paramétrique est égal par définition à

l'intégrale d'une densité de probabilité sur une certaine région d'acceptabilité. Ainsi, maximiser le rendement revient à maximiser le volume délimité par l'enveloppe de la densité et la région d'acceptabilité. Cet objectif peut être atteint indirectement en centrant la densité de probabilité sur la région d'acceptabilité (cf. Figure 3.13), d'où le nom de ces méthodes dites de « design centering » [73].

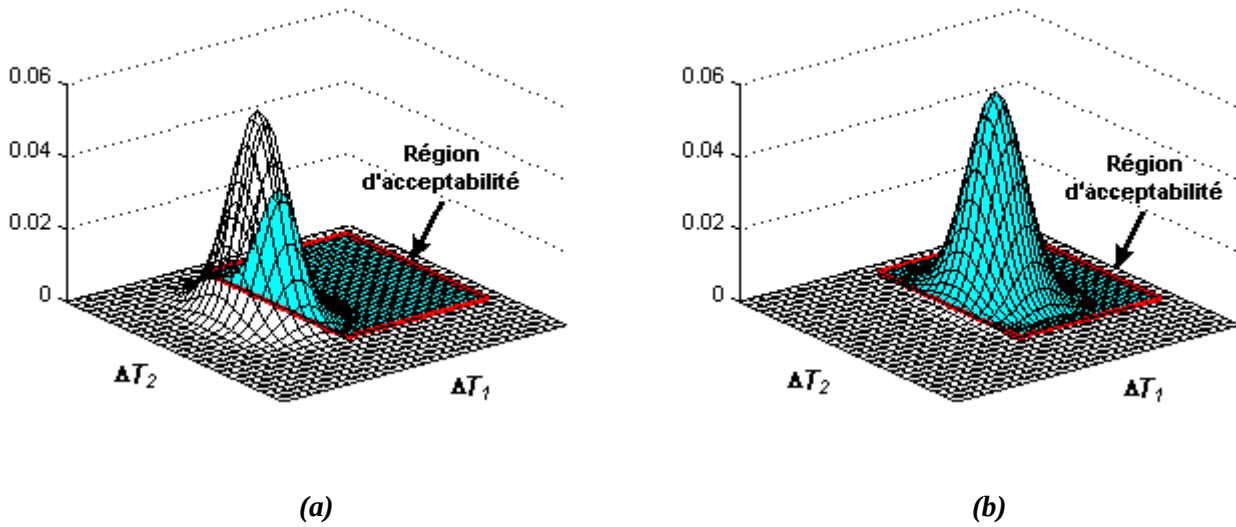


Figure 3.13 : Maximisation du rendement (volume bleu) en centrant la densité de probabilité sur la région d'acceptabilité (rectangle rouge)

Deux approches sont envisageables comme l'illustre la Figure 3.14 : centrer la distribution des performances sur la région d'acceptabilité A_Y ou bien centrer la distribution des variations technologiques sur la région d'acceptabilité $A_{\Delta T}(C)$. Dans le premier cas de figure, la région d'acceptabilité A_Y est facilement caractérisable mais la distribution des performances n'est pas connue. Dans le second cas de figure, la distribution des variations technologiques est connue mais cette fois-ci c'est la région d'acceptabilité $A_{\Delta T}(C)$ qui est inconnue.

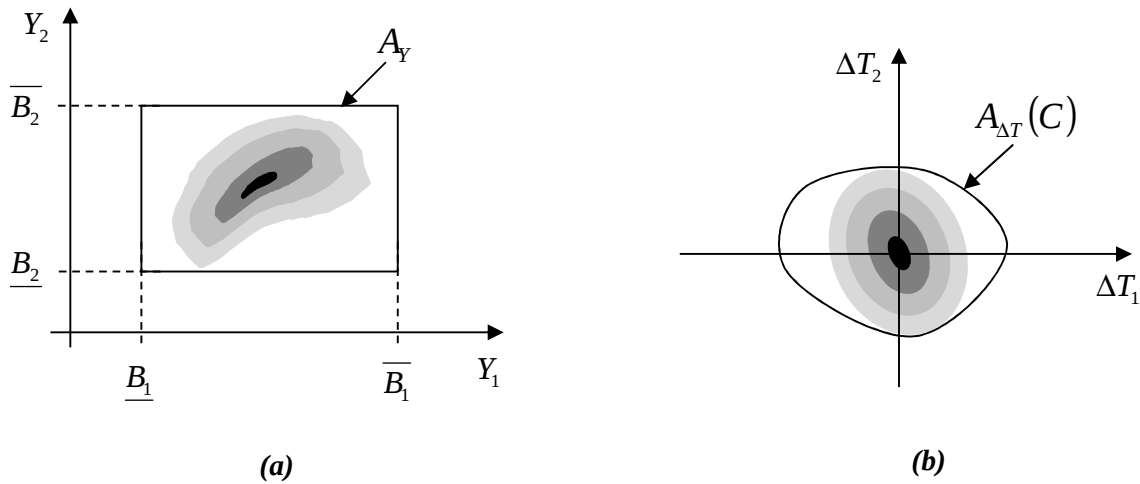


Figure 3.14 : Centrage de la distribution des performances sur la région d'acceptabilité A_Y (a) et centrage de la distribution des variations technologiques sur la région d'acceptabilité $A_{AT}(C)$ (b)

Une première famille de méthodes, destinée aux circuits discrets, consiste à réaliser une approximation polyédrique [51, 52, 74] ou ellipsoïdale [53, 54] de la région d'acceptabilité A_C des paramètres de conception. Parmi ces méthodes, certaines choisissent simplement le centre de ces approximations comme point de conception robuste [53, 54]. D'autres tiennent compte de la forme de la distribution. Dans le cas d'une loi normale, les courbes d'isodensité de la densité de probabilité forment des ellipses qui sont de plus en plus grandes au fur et à mesure qu'on s'éloigne de la valeur moyenne. Le rendement maximal sera donc obtenu en inscrivant la plus grande de ces ellipses dans l'approximation de la région d'acceptabilité ; le centre de l'ellipse correspond alors au point de conception le plus robuste [51, 52, 74]. Cependant, ces méthodes ne sont pas adaptées si la région d'acceptabilité n'est pas convexe. De plus, dans le cas de l'approximation polyédrique [51] appelée aussi « simplicial approximation », le coût calculatoire augmente rapidement avec le nombre de paramètres. Enfin, ces méthodes ne sont applicables qu'aux circuits discrets.

Une deuxième famille de méthodes, adaptée aux circuits intégrés, consiste à rechercher un dimensionnement robuste en maximisant les distances pire-cas entre le centre de la distribution et la frontière de la région d'acceptabilité (cf. Figure 3.15). Le nombre de distances pire-cas dépend du nombre de contraintes ; elles peuvent être calculées en ayant recours à un algorithme d'optimisation [75] ou en linéarisant les contraintes [76, 77]. Les valeurs des paramètres de conception qui maximisent les distances pire-cas sont ensuite obtenues en résolvant un problème d'optimisation multi-objectif. En termes de complexité, l'approche présentée dans [75] présente un coût de calcul qui n'augmente que linéairement avec le nombre de paramètres de conception mais fait appel à des simulations électriques à chaque itération pour évaluer les distances pire-cas et leur gradient. Les mé-

thodes [76, 77] ont recours quant à elles à des approximations linéaires des performances afin d'alléger le coût calculatoire.

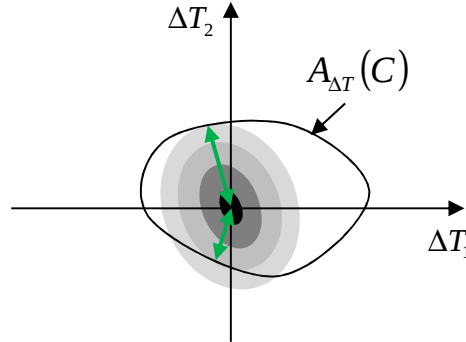


Figure 3.15 : Illustration des distances pire-cas (flèches en vert)

En termes de précision, les méthodes de « design centering » sont basées sur une approximation plus ou moins précise de la région d'acceptabilité ; elles ne conduisent donc qu'à une solution approchée du rendement maximum. Cependant, en formulant le problème d'optimisation du rendement sous forme déterministe, les méthodes de « design centering » convergent plus rapidement que les méthodes statistiques.

3.3.4 Optimisation robuste - Optimisation des performances pire-cas

Le problème de conception robuste, tel que nous l'avons défini plus haut, peut s'exprimer sous la forme d'un problème d'optimisation dont la résolution conduit à un jeu de valeurs pour les paramètres de conception qui minimise une certaine fonction de coût f et qui garantit que les spécifications sur des performances g_l sont respectées pour toutes les dispersions paramétriques appartenant à la région de tolérance $RT_{\Delta T}$. Le problème de conception robuste peut donc s'écrire sous la forme générale suivante :

$$\begin{aligned}
 &\underset{\mathbf{C}}{\text{minimiser}} && f(\mathbf{C}, \Delta \mathbf{T}) \\
 &\text{tel que} && g_l(\mathbf{C}, \Delta \mathbf{T}) < 0, \quad l \in \{1, \dots, p\} \\
 &&& \mathbf{C} \in D \\
 &&& \forall \Delta \mathbf{T} \in RT_{\Delta T}
 \end{aligned} \tag{3.21}$$

où

$\mathbf{C}$ est le vecteur des paramètres de conception appartenant au domaine de conception D ,

$\Delta \mathbf{T}$ est le vecteur des variations technologiques appartenant à la région de tolérance $RT_{\Delta T}$,

f est la fonction de coût à minimiser,

g_l sont les performances qui doivent respecter les spécifications définies par des inégalités.

La difficulté majeure dans ce problème d'optimisation est liée à la contrainte « $\forall \Delta T \in RT_{\Delta T}$ », c'est-à-dire le « quelles que soient les fluctuations paramétriques ΔT appartenant à la région de tolérance $RT_{\Delta T}$ » dans la formulation du problème de conception robuste. La région de tolérance $RT_{\Delta T}$ étant un ensemble infini, la contrainte « $\forall \Delta T \in RT_{\Delta T}$ » peut donc être vue comme une infinité de contraintes qui doivent être vérifiées. Une solution pour s'affranchir de cette difficulté consiste à reformuler les contraintes en faisant intervenir la borne supérieure des performances. En effet, si la borne supérieure des performances sur $RT_{\Delta T}$ vérifie les contraintes, alors ces dernières sont vérifiées quelles que soient les valeurs de $RT_{\Delta T}$. Le problème (3.21) peut alors se reformuler sous la forme suivante, également appelée optimisation robuste [78] :

$$\begin{aligned} & \underset{\mathbf{C}}{\text{minimiser}} && \sup_{\Delta T \in RT_{\Delta T}} f(\mathbf{C}, \Delta T) \\ & \text{tel que} && \sup_{\Delta T \in RT_{\Delta T}} g_l(\mathbf{C}, \Delta T) < 0, \quad l \in \{1, \dots, p\} \\ & && \mathbf{C} \in D \end{aligned} \quad (3.22)$$

Grâce à cette reformulation, l'ensemble infini des contraintes a été remplacé par une seule contrainte « pire-cas » sur les performances (à ne pas confondre avec l'analyse pire-cas vue précédemment), facilitant ainsi la résolution du problème de conception robuste.

Différentes méthodes ont été proposées pour résoudre ce problème d'optimisation robuste. La méthode présentée dans [79] résout directement le problème (3.21) grâce à des techniques de programmation infinie combinées à l'algorithme du recuit-simulé. Dans [80], les auteurs représentent la région de tolérance $RT_{\Delta T}$ sous forme elliptique et approximent les performances par des modèles posynomiaux ; le problème (3.22) est alors résolu par un algorithme de programmation géométrique robuste [81]. Une autre approche consiste à résoudre le problème (3.22) en optimisant les performances pire-cas : à chaque itération, les bornes supérieures des performances sont approximées par les performances pire-cas extraites avec APEX [82] ou une analyse de Monte Carlo [83]. Enfin, dans [84], les auteurs ont recours à l'arithmétique affine, qui est une extension de l'arithmétique des intervalles présentée dans l'Annexe C, afin d'estimer les performances pire-cas et mettent en œuvre un algorithme d'optimisation par intervalles de type « Branch and Bound » (cf. §6.3.2b) pour résoudre le problème (3.22).

A la différence des méthodes d'optimisation du rendement où la seule grandeur optimisée est le rendement, les méthodes d'optimisation robuste permettent d'optimiser une ou plusieurs perfor-

mances et en plus de garantir un rendement élevé en vérifiant que les variations extrêmes sont respectées. Les méthodes d'optimisation robuste offrent donc une plus grande flexibilité.

3.3.5 Optimisation par les indices de capabilité

Une dernière approche indirecte pour traiter la conception robuste est dérivée de la méthode d'amélioration de la qualité mise au point par G. Taguchi. Cette approche est fondée sur deux critères de mesure de la variabilité appelés indices de capabilité [85] :

$$C_p = \frac{\bar{B} - \underline{B}}{6\sigma_Y} \quad (3.23)$$

$$C_{pk} = \min \left\{ \frac{\bar{B} - \mu_Y}{3\sigma_Y}, \frac{\mu_Y - \underline{B}}{3\sigma_Y} \right\} \quad (3.24)$$

où $\underline{B}$ et $\bar{B}$ représentent respectivement les spécifications inférieure et supérieure d'une performance Y , tandis que μ_Y et σ_Y sont respectivement sa moyenne et son écart type. L'indice C_p quantifie la variabilité de la performance : plus cet indice est grand, moins la performance varie (cf. Figure 3.16(a)). Quant à l'indice C_{pk} (cf. Figure 3.16(b)), il permet de caractériser le centrage de la performance : si la performance est correctement centrée, alors $\mu_Y = (\underline{B} + \bar{B})/2$ et $C_{pk} = C_p$.

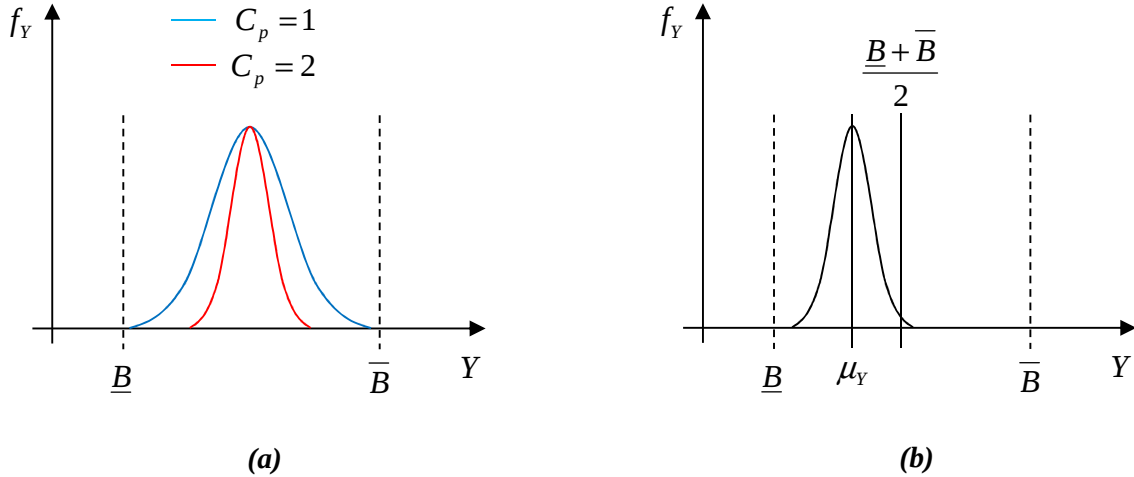


Figure 3.16 : Illustration des indices C_p (a) et C_{pk} (b)

En combinant ces deux indices, on peut définir pour chaque performance l une mesure de la robustesse :

$$\phi_l^L = C_{p,l} + \lambda \frac{(\underline{B} + \overline{B})/2 - \mu_Y}{3\sigma_Y} \quad (3.25)$$

$$\phi_l^U = C_{p,l} + \lambda \frac{\mu_Y - (\underline{B} + \overline{B})/2}{3\sigma_Y} \quad (3.26)$$

où $0 \leq \lambda \leq 1$ est un paramètre qui peut être modifié au cours de l'optimisation. La conception robuste par les indices de capabilité consiste donc à déterminer les paramètres de conception C qui maximisent la fonction de coût suivante :

$$\max_{C \in D} \left(\min_{l=1,\dots,p} \{ \phi_l^L, \phi_l^U \} \right) \quad (3.27)$$

où D est le domaine de conception et p le nombre de performances. Si $\lambda = 0$, alors la fonction de coût représente l'indice C_p , c'est donc la variabilité des performances que l'on cherche à minimiser. Il s'agit de la première étape de l'approche de Taguchi. A l'opposé, si $\lambda = 1$, la fonction de coût correspond à l'indice C_{pk} , on cherche donc à centrer les performances par rapport à leurs spécifications. Il s'agit de la seconde étape de l'approche de Taguchi. L'intérêt principal de cette méthode est de pouvoir contrôler indépendamment la variabilité des performances et leur centrage [85, 86]. Cependant, l'utilisation des indices de capabilité est fondée sur l'hypothèse de normalité des performances, qui est loin d'être vérifiée en pratique.

3.3.6 Bilan

Les caractéristiques des méthodes de conception robuste que nous avons présentées sont résumées dans le Tableau 3.2. Elles se différencient par leur précision, leur complexité et les conditions qu'elles imposent sur la distribution des variations. Les méthodes statistiques d'optimisation du rendement sont précises mais très lourdes en termes de calcul. Les méthodes de « Design Centring » convergent plus rapidement, mais le résultat obtenu dépend de l'approximation de la région d'acceptabilité. Les méthodes d'optimisation robuste offrent plus de flexibilité en permettant d'optimiser les performances tout en assurant un rendement paramétrique élevé. Cependant, elles requièrent des techniques efficaces pour évaluer les performances pire-cas. Enfin, l'optimisation par indices de capabilité réalise un dimensionnement robuste en se concentrant sur le centrage des performances et la réduction de la variabilité, mais suppose une distribution normale des performances.

Tableau 3.2 : Méthodes de conception robuste

Méthodes	Avantages	Inconvénients
Optimisation du rendement (méthodes statistiques)	- pas de contraintes sur la forme de la distribution des variations technologiques - estimation précise du rendement	- coût calculatoire très élevé
Optimisation du rendement (« Design Centering »)	- formulation indirecte du problème d'optimisation pour assurer une meilleure convergence	- la précision de l'optimum dépend de l'approximation de la région d'acceptabilité
Optimisation robuste	- optimisation des performances pire-cas en garantissant un rendement élevé	- estimation des performances pire-cas
Indices de capacité	- centrage des performances et réduction de leur variabilité	- hypothèse de normalité des performances

3.4 Problématique de la thèse

La prise en compte de la variabilité dans le flot de conception augmente considérablement le coût calculatoire. Pour réduire ce coût, les méthodes de conception robuste que nous avons présentées reposent donc sur des algorithmes d'optimisation rapides mais locaux. L'optimum global est alors généralement atteint en deux étapes. La première étape consiste à réaliser une conception nominale à partir d'algorithmes d'optimisation globaux : les performances des circuits sont optimisées sans tenir compte des variations. De nombreux logiciels commerciaux ou académiques sont à présent disponibles [65] pour réaliser la conception nominale. Après cette première étape, les méthodes de conception robuste basées sur des algorithmes d'optimisation locaux sont appliquées : l'optimum de la conception nominale sert alors de point de départ aux méthodes de conception robuste. Le point faible de cette approche est que l'optimum nominal se trouve souvent à la limite des spécifications : la moindre variation paramétrique provoque alors la violation des spécifications, ce qui dégrade le rendement paramétrique. L'optimum de la conception nominale constitue donc un maximum local en termes de rendement ; les méthodes de conception robuste qui font appel à des algorithmes d'optimisation locaux et utilisent l'optimum de la conception nominale pour s'initialiser restent alors bloquées sur ce maximum local du rendement. Afin d'identifier l'optimum global robuste, il est donc important de prendre en compte la variabilité dès le début du flot de conception et non uniquement à la fin [79, 86, 87].

Par conséquent, le développement de méthodes de conception robuste efficaces nécessite la mise en place des points suivants :

- une méthode d'analyse de la variabilité afin d'appréhender l'impact des dispersions paramétriques sur les performances du circuit,
- un algorithme d'optimisation globale mis en place dès le début du flot de conception et qui aboutit à un dimensionnement automatisé, optimal, ainsi que robuste en appliquant la méthode d'analyse de la variabilité,
- déterminer les approches permettant d'atteindre le meilleur compromis précision/complexité à chaque étape de l'implémentation de la méthode de conception robuste afin d'accélérer sa convergence.

Dans ce travail de thèse, une nouvelle méthode de conception robuste a été développée en tenant compte des considérations précédentes. Cette méthode est basée sur l'optimisation des performances pire-cas présentée dans la partie §3.3.4. Ses principales caractéristiques sont les suivantes :

- les performances pire-cas sont estimées en ayant recours à une modélisation polynomiale des performances et au développement limité de Cornish-Fisher,
- l'optimisation est réalisée en utilisant un algorithme par intervalles qui garantit que l'optimum global sera atteint.

Une attention particulière a été portée sur la réduction du coût calculatoire. Ainsi, la modélisation polynomiale des performances fait appel à la théorie des plans d'expériences pour limiter le nombre de simulations à réaliser, tandis que des techniques de réduction d'intervalles sont utilisées dans l'algorithme par intervalles pour accélérer sa convergence. Ces différents points sont expliqués dans les chapitres suivants.

Chapitre 4 : Analyse de la variabilité à partir des modèles polynomiaux et de l'approximation de Cornish-Fisher

Ce chapitre présente une nouvelle méthode permettant d'analyser la dispersion des performances de circuits analogiques en fonction des variations des paramètres technologiques, pour des paramètres de conception fixés. Après une présentation générale de la méthode, les deux points clés de cette dernière, à savoir l'approximation des performances à partir des plans d'expériences et l'estimation des bornes de variation avec le développement limité de Cornish-Fisher, seront ensuite détaillés. D'après les résultats obtenus sur des circuits analogiques, la méthode proposée se révèle aussi précise que l'analyse de Monte Carlo, tout en étant plus rapide.

4.1 Présentation de la méthode d'analyse de la variabilité

Dans le Chapitre 3, nous avons passé en revue plusieurs méthodes d'analyse de la variabilité dédiées aux circuits analogiques. Leur objectif est de caractériser les variations des performances à partir de mesures statistiques comme leur densité de probabilité ou le rendement paramétrique. La méthode d'analyse de variabilité que nous allons présenter a pour objectif d'estimer les valeurs pire-cas des performances, c'est-à-dire les quantiles extrêmes de sa distribution comme par exemple les 1^{er} et 99^{ème} centiles. Cette méthode est destinée à être appelée à chaque itération d'un algorithme d'optimisation robuste ; son coût calculatoire doit donc être faible tout en préservant une précision suffisante. Elle consiste dans un premier temps à approximer les performances des circuits par des modèles plus simples appelés méta-modèles. Ces méta-modèles permettent de décrire la relation qui existe entre les variations des paramètres technologiques et la performance étudiée, pour des valeurs de paramètres de conception données. Ensuite, une fois ces méta-modèles construits, en l'occurrence des polynômes, les cumulants des performances sont calculés à partir des variations des paramètres technologiques et des coefficients des méta-modèles. Enfin les valeurs pire-cas des performances peuvent être estimées en appliquant le développement limité de Cornish-Fisher. Les différentes étapes de la méthode sont illustrées sur la Figure 4.1.

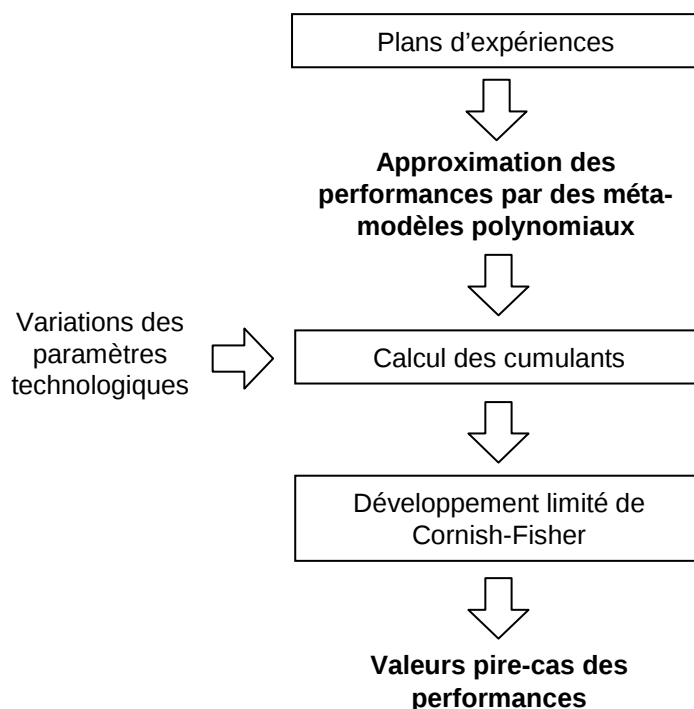


Figure 4.1 : Méthode d'analyse de la variabilité proposée

4.2 Construction d'un modèle polynomial des performances avec les plans d'expériences

Dans cette partie, nous allons présenter la première étape de notre méthode d'analyse de la variabilité, à savoir l'approximation des performances des circuits analogiques par des modèles polynomiaux en utilisant les plans d'expériences. Les principes de base de l'approximation par des méta-modèles seront d'abord introduits, avant d'expliquer plus en détail la construction des modèles polynomiaux et les plans d'expériences associés.

4.2.1 Intérêt des méta-modèles

Grâce aux progrès de l'informatique, de plus en plus de phénomènes physiques qui nécessiteraient de lourdes expérimentations pour être analysés, sont étudiés via des modèles numériques. Un modèle numérique peut être vu comme un programme informatique générant une réponse en sortie qui dépend de plusieurs paramètres en entrée (cf. Figure 4.2). Cependant, malgré l'augmentation de la puissance de calcul des ordinateurs, la complexité de certains simulateurs informatiques est telle que l'impact des différents paramètres en jeu reste difficile à appréhender. De plus, les temps de simulation restent très longs, rendant impossible l'exploration complète du domaine de valeurs des

paramètres d'entrée et leur emploi dans des algorithmes d'optimisation qui requièrent plusieurs itérations de simulations. Pour réduire les temps de calcul, une approche couramment utilisée consiste à approximer la réponse du simulateur par un modèle plus simple, moins précis que le simulateur, mais moins long à simuler. Ces méta-modèles, c'est-à-dire « modèles de modèles » [88], peuvent ainsi efficacement remplacer les simulateurs complexes dans plusieurs cas de figure. Tout d'abord, ces méta-modèles donnent une meilleure compréhension de la relation qui unit la réponse du simulateur avec ses paramètres d'entrée, permettant ainsi d'identifier les paramètres d'entrée qui ont une réelle influence sur la réponse du simulateur. De plus, ils permettent de prédire à moindre coût la forme globale de la réponse du simulateur sur l'espace des valeurs des paramètres d'entrée. Enfin, leur faible coût de calcul rend possible leur intégration dans des algorithmes d'optimisation.

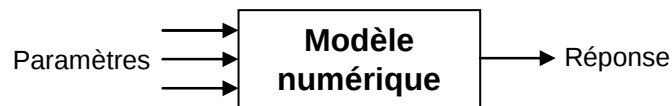


Figure 4.2 : Représentation d'un modèle numérique

Les simulateurs utilisés pour la conception de circuits analogiques sont des simulateurs électriques comme Spice [37], Eldo [38] ou Spectre [39] qui s'appuient sur des modèles de transistors d'autant plus complexes que les dimensions de ces derniers diminuent. Les variables d'entrée de ces simulateurs sont généralement les paramètres de conception du circuit, les paramètres technologiques et les paramètres environnementaux, tandis que leurs réponses correspondent aux performances du circuit (cf. §3.1.1). Dans l'approche présentée, les performances extraites d'un simulateur électrique sont approximées par des méta-modèles plus simples qui vont permettre de caractériser les relations entre les paramètres d'entrée et les performances du circuit. A partir de ce méta-modèle, il sera alors possible d'analyser l'impact des variations des paramètres technologiques sur les performances, pour un jeu de valeurs donné des paramètres de conception et des paramètres environnementaux.

4.2.2 Les plans d'expériences

Les méta-modèles doivent être aussi précis que possible pour prédire au mieux la réponse du simulateur. Cependant, augmenter la précision des méta-modèles se traduit généralement par une augmentation du coût calculatoire. La construction des méta-modèles est donc réalisée en conciliant deux contraintes : obtenir des méta-modèles suffisamment précis, tout en limitant le coût des calculs pour les construire. Pour atteindre ce compromis, une approche possible consiste à utiliser la théorie

des plans d'expériences associée à des méthodes de construction de surface de réponses. Cette approche se déroule en cinq étapes :

- choisir un type de méta-modèle,
- déterminer le plan d'expériences conduisant au méta-modèle le plus précis pour un coût calculatoire minimum,
- évaluer les valeurs de la réponse aux points fixés par le plan d'expériences en faisant appel au simulateur,
- estimer les coefficients du méta-modèle à partir des valeurs de la réponse extraites des simulations (maximum de vraisemblance, régression linéaire, etc.),
- valider le méta-modèle en testant sa précision.

a) Expériences physiques et numériques

Les plans d'expériences ont été développés à l'origine pour l'étude de phénomènes physiques. Ce n'est que plus tard qu'ils ont été appliqués à la modélisation de simulateurs informatiques. Avant de détailler chacune des étapes ci-dessus, il est donc important de préciser les différences entre les expériences physiques et les expériences numériques dans la mesure où ces différences conduisent à des choix de méthodes distincts lors de la construction d'un méta-modèle. Dans le cas d'expériences physiques, chaque expérimentation est entachée d'une incertitude appelée erreur expérimentale. En effet, si on réalise plusieurs fois la même expérience physique, on n'obtient pas exactement le même résultat, du fait de l'instabilité des instruments, des fluctuations des conditions expérimentales ou des erreurs de lecture. Dans le cas d'un simulateur informatique, les réponses sont déterministes, il n'y a pas d'erreur expérimentale : deux simulations avec les mêmes paramètres d'entrée donnent la même réponse. De plus, les paramètres d'entrée d'un modèle numérique sont en général plus nombreux, tandis que sa complexité rend sa réponse souvent discontinue [89].

b) Les méta-modèles

Propriétés

Le choix du type de méta-modèle dépend essentiellement de deux propriétés importantes du modèle numérique à approximer : sa dimension (nombre de paramètres en entrée) et son degré de non-linéarité. De plus, pour déterminer le méta-modèle le plus adapté, il est nécessaire d'avoir en tête les principaux critères permettant de les comparer entre eux, à savoir [90] :

- la précision de leur prédiction,

- le coût calculatoire pour les construire,
- la généralité, c'est-à-dire la faculté de pouvoir approximer tout type de réponses,
- la lisibilité (la forme du méta-modèle permet-elle d'extraire aisément les contributions des différents paramètres ?),
- la simplicité de la méthode d'implémentation pour générer ces méta-modèles.

Modélisation

La relation entre une réponse Y et un vecteur de paramètres d'entrée X peut s'exprimer par l'intermédiaire d'un méta-modèle $\hat{f}$:

$$Y = \hat{f}(X) + \varepsilon \quad (4.1)$$

où ε est un terme d'erreur, généralement modélisé par une variable aléatoire d'espérance nulle et d'écart type σ . Dans le cas d'expériences physiques, ce terme d'erreur englobe à la fois l'erreur expérimentale dont la nature est purement aléatoire, ainsi que le manque d'ajustement entre le méta-modèle et le modèle réel du phénomène étudié. Pour des expériences numériques, il n'y a pas d'erreur expérimentale comme nous l'avons vu. Le terme d'erreur ε représente alors uniquement le manque d'ajustement. La modélisation du manque d'ajustement par une variable aléatoire reste cependant valide dans la mesure où « l'objet de l'étude n'est pas la réponse du simulateur, mais le phénomène simulé, qui peut être considéré comme la réponse du simulateur plus une erreur aléatoire due aux simplifications du modèle mathématique » [89].

Types de méta-modèles

Parmi les différents types de méta-modèles possibles, les modèles basés sur des polynômes font partie des méta-modèles les plus utilisés. D'une part, parce qu'ils sont facilement interprétables et que leur coût de construction est faible. D'autre part, parce qu'ils permettent de réaliser une analyse de sensibilité permettant entre autres d'identifier les paramètres influents. Cependant, en cas de réponses extrêmement non-linéaires à approximer, les modèles polynomiaux d'ordre trois ou plus peuvent se révéler instables (incertitude élevée sur l'estimation des coefficients), et de plus, le nombre d'expériences à réaliser pour construire de tels modèles croît rapidement avec le nombre de paramètres. Malgré tout, même si la réponse a globalement un comportement complexe, un simple modèle linéaire ou quadratique peut fournir localement une approximation satisfaisante. Pour des réponses non-linéaires, l'approche par krigeage [91] donne de meilleurs résultats que les modèles polynomiaux. Il s'agit d'une méthode d'interpolation qui consiste à approximer la réponse par un processus gaussien dont la moyenne représente la forme générale de la réponse et la covariance sa

régularité. En tant qu'interpolateur, le krigeage est particulièrement intéressant pour les expériences numériques puisqu'il passe par tous les points simulés. Cependant, l'estimation de ses paramètres nécessite de résoudre un problème d'optimisation globale non-linéaire, ce qui alourdit le temps de calcul. De plus, la construction des modèles de krigeage fait appel à une fonction de corrélation fixée à priori, mais rien ne permet d'aider l'utilisateur dans le choix de cette fonction [89]. Parmi les autres méta-modèles non-linéaires, les fonctions à base radiale constituent également une alternative intéressante puisqu'il s'agit de méthodes d'interpolation comme le krigeage, mais moins coûteuses en calcul [92]. D'autres types de méta-modèles non-linéaires sont envisageables comme les réseaux de neurones [93], les modèles posynomiaux [94, 95], les machines à vecteurs de support [96] ou la régression par splines multiples et adaptives (MARS). Enfin, lorsque le nombre de paramètres est élevé, des méthodes combinant modélisation et réduction du coût calculatoire ont été développées. On peut citer par exemple les modèles de poursuite de projection [97, 98] qui approximent la réponse par une somme de fonctions unidimensionnelles, le développement en polynôme de chaos [99, 100] si les paramètres sont des variables aléatoires ou encore les techniques de réduction d'ordre des modèles [101, 102]. Dans [90], différents modèles ont été comparés suivant les critères énoncés plus haut et ont été repris dans le Tableau 4.1. Il ressort de cette étude que les modèles polynomiaux sont bien adaptés pour les modèles numériques qui ont peu de paramètres en entrée et sont faiblement non-linéaires. Nous avons donc retenu ce type de méta-modèle pour approximer localement les performances de circuits analogiques basiques (moins de vingt transistors, fonctionnement continu).

Tableau 4.1 : Comparaison de différents types de méta-modèles en fonction de la dimension (nombre de paramètres) et du degré de non-linéarité du modèle numérique à approximer d'après [90]

		Non-linéarité du modèle numérique	
		Faible	Elevée
Dimension du modèle numérique	Faible	Polynômes	Fonctions à base radiale
	Elevée	Krigeage	Fonctions à base radiale

c) Les plans d'expériences classiques et numériques

Définitions

La méthode dite des plans d'expériences peut être définie au sens large comme la mise en place d'une campagne d'essais afin d'extraire un maximum de renseignements à partir d'un minimum d'expérimentations pour répondre à un problème donné [103, 104]. Un plan d'expériences se com-

pose donc d'un certain nombre d'essais, chaque essai correspondant à une combinaison particulière de valeurs des paramètres d'entrée. Dans la théorie des plans d'expériences, les paramètres d'entrée et la grandeur étudiée sont respectivement appelés « facteurs » et « réponse », tandis que le terme « niveau » est utilisé pour désigner les valeurs auxquelles ces facteurs sont fixés. Lors de la mise en place d'un plan d'expériences, on limite l'étude de chaque facteur à un intervalle de valeurs appelé domaine de variation ; le produit cartésien des domaines de variation de chaque facteur forme le domaine d'étude. Par la suite, ce domaine d'étude correspondra au domaine de validité du méta-modèle créé à partir du plan d'expériences. De plus, afin d'avoir des représentations matricielles plus générales, mais également pouvoir comparer directement l'effet des facteurs, une transformation est appliquée aux facteurs d'origine de telle sorte que le domaine de variation des nouveaux facteurs corresponde à l'intervalle normalisé $[-1, +1]$. Cette transformation revient à effectuer le changement d'origine et d'unité suivant :

$$\begin{aligned} [\underline{X}_i, \overline{X}_i] &\rightarrow [-1, +1] \\ X_i &\rightarrow X_{in} = \left(X_i - \frac{\underline{X}_i + \overline{X}_i}{2} \right) \times \frac{1}{\frac{\overline{X}_i - \underline{X}_i}{2}} \end{aligned} \quad (4.2)$$

où X_i est le facteur d'origine, $\underline{X}_i$ et $\overline{X}_i$ les bornes de son domaine de variation, et X_{in} le nouveau facteur normalisé. Enfin, il est possible de définir un plan d'expériences sous forme matricielle par un tableau appelé matrice des essais donnant les niveaux des facteurs (cf. Figure 4.3(a)). S'il y a au plus trois facteurs, une représentation graphique des essais (cf. Figure 4.3(b)) est également possible en représentant chaque facteur sur un axe gradué et orienté dans l'espace, et en représentant chaque essai par un point dont les coordonnées sont définies par la matrice des essais.

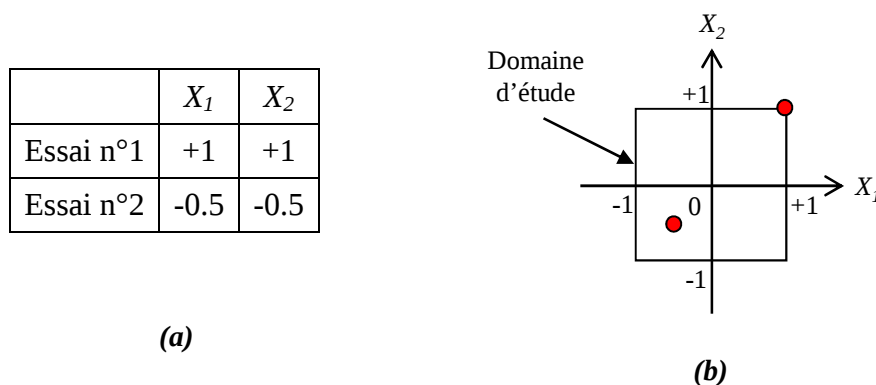


Figure 4.3 : Représentations matricielle (a) et graphique (b) d'un plan d'expériences comportant deux facteurs et deux essais

Plans d'expériences classiques

Historiquement les plans d'expériences furent introduits dans les années 1920 par R.A. Fisher et d'abord appliqués à l'agronomie [105]. Dans les années 1950, ils furent étendus au domaine médical, ainsi qu'au secteur industriel afin d'établir une relation entre une grandeur physique et plusieurs facteurs permettant ainsi de déterminer l'influence respective de ces facteurs ou d'optimiser un procédé (méthodologie de « Response Surface Modeling » [106], approche qualité de Taguchi [107] par exemple). Lorsque la grandeur étudiée concerne un phénomène physique (rendement d'une parcelle agricole, efficacité d'un médicament, robustesse d'un produit manufacturier), on parle de plans d'expériences classiques. Dans le cas des plans d'expériences classiques, la relation entre la grandeur d'intérêt et les paramètres d'entrée est généralement représentée par un modèle linéaire ou quadratique. L'intérêt des plans d'expériences est alors de définir, pour un type de modèle donné, le nombre minimum et l'emplacement des essais dans le domaine d'étude permettant d'estimer les coefficients du modèle avec la meilleure précision possible. Les points expérimentaux de ce type de plan sont donc généralement situés sur le bord du domaine expérimental afin de minimiser l'erreur sur les coefficients du modèle obtenus par régression linéaire. Dans le cas des plans d'expériences physiques, chaque essai est entaché d'une erreur expérimentale. La régression linéaire basée sur le critère des moindres carrés a donc pour résultat de lisser cette erreur expérimentale.

Plans d'expériences numériques

Dans le cas d'un modèle numérique, les réponses sont déterministes, il n'y a pas d'erreur expérimentale. C'est pour cette raison que le modèle de krigeage, basé sur une méthode d'interpolation, et qui passe donc par tous les points simulés, est particulièrement bien adapté aux expériences numériques. Les plans d'expériences numériques mis en œuvre pour ce type de modèle (hypercubes latins, tableaux orthogonaux, etc.) sont constitués de points de simulation répartis de façon uniforme dans le domaine d'étude afin de détecter d'éventuelles irrégularités à l'intérieur du domaine.

Comme nous l'avons vu, les méta-modèles envisagés pour approximer les performances de circuits analogiques sont des modèles polynomiaux. Ce type de méta-modèle est généralement construit à partir des plans d'expériences classiques qui sont traditionnellement appliqués à des expériences physiques. Dans le cadre de cette thèse, nous allons donc appliquer ces plans d'expériences classiques à des simulations numériques.

d) L'évaluation des valeurs de la réponse

Une simulation est réalisée pour chaque essai du plan d'expériences afin de calculer la valeur de la réponse correspondante. Il y a donc autant de simulations réalisées que d'essais dans le plan d'expériences. Dans certains cas, il peut être intéressant d'appliquer une transformation ($\log()$,

exp(), etc.) sur les valeurs de la réponse du simulateur afin d'améliorer la qualité de l'approximation.

e) L'estimation des paramètres du méta-modèle

Une fois que tous les essais du plan d'expériences ont été simulés et les valeurs de la réponse extraites, les paramètres du méta-modèle peuvent alors être estimés. Suivant le type de méta-modèle, différentes méthodes d'estimation existent :

- la régression multilinéaire basée sur l'estimateur des moindres carrés (méta-modèles polynomiaux)
- l'estimateur du maximum de vraisemblance (modèles de krigeage)
- la rétropropagation (réseaux de neurones)
- l'entropie (arbres de décision)

f) La validation du méta-modèle

Méthodes statistiques

La dernière étape dans la construction d'un méta-modèle consiste à tester sa précision par rapport au modèle numérique qu'il approxime. Dans le cas des expériences physiques, la mise en place de procédures, comme la répétition d'essais au centre du domaine d'étude pour estimer l'erreur expérimentale, associées à des tests statistiques (analyse de la variance) permet d'obtenir un certain nombre d'informations pour valider le modèle. Cependant, pour des expériences numériques, il n'y a pas d'erreur expérimentale et la répétition d'essais en un point est sans intérêt puisque ceux-ci donnent le même résultat.

Validation croisée

Dans le cas des expériences numériques, les méthodes de validation croisée semblent bien adaptées. Elles consistent à construire des méta-modèles sur des sous-ensembles disjoints d'essais du plan d'expériences, puis à estimer la précision de chacun des méta-modèles à partir des essais qui n'ont pas été utilisés pour sa construction. Un cas particulier de validation croisée est le « leave-one-out » où les sous-ensembles utilisés correspondent au plan d'expériences initial privé d'un essai qui est différent à chaque fois. L'algorithme du « leave-one-out » est illustré sur la Figure 4.4. Plus la variance de validation croisée *VVC* est faible, plus l'erreur de prédiction du méta-modèle sera faible.

E est l'ensemble des essais du plan d'expériences : $E = \{e_1, \dots, e_n\}$

f est le modèle numérique à approximer

$VVC = 0$

Pour i de 1 à n

Construire un méta-modèle $\hat{f}_i$ de f à partir de l'ensemble des essais privés de $e_i : E \setminus \{e_i\}$

$VVC = VVC + (f(e_i) - \hat{f}_i(e_i))^2$

Fin Pour

Figure 4.4 : Algorithme de validation croisée « leave-one-out »

Critères proposés

Cependant, lorsqu'il s'agit des plans d'expériences classiques pour approximer des méta-modèles polynomiaux, le fait de retirer un essai du plan détruit sa structure, il ne possède alors plus les propriétés nécessaires pour fournir une bonne approximation des coefficients du modèle [89]. Il faut donc avoir recours à des essais supplémentaires, différents de ceux utilisés pour construire le modèle. La précision du méta-modèle est ainsi testée à partir d'autres critères de validation comme l'erreur quadratique moyenne $RMSE$ (« Root Mean Square Error » en anglais), l'erreur absolue maximum MAX ou encore le coefficient R^2 [92] :

$$RMSE = \sqrt{\frac{\sum_{i=1}^N (y_i - \hat{y}_i)^2}{N}} \quad (4.3)$$

$$MAX = \max |y_i - \hat{y}_i|, \quad i = 1, \dots, N \quad (4.4)$$

$$R^2 = 1 - \frac{\sum_{i=1}^N (y_i - \hat{y}_i)^2}{\sum_{i=1}^N (y_i - \bar{y})^2} \quad (4.5)$$

où $\hat{y}_i$ est la valeur prédite par le méta-modèle, y_i la valeur simulée (réelle), $\bar{y}$ la moyenne des valeurs réelles et N le nombre d'essais supplémentaires. Les critères $RMSE$ et R^2 permettent de juger de la précision globale du méta-modèle, tandis que le critère MAX caractérise sa précision locale. Dans la suite de cette thèse, nous utiliserons ces critères pour valider les méta-modèles polynomiaux générés à partir des plans d'expériences.

4.2.3 Modèles polynomiaux d'ordre 1 et 2

Etant donné leur construction aisée et la qualité de leurs prédictions locales, les méta-modèles polynomiaux ont été retenus pour approximer les performances des circuits analogiques où une dizaine de paramètres peuvent être en jeu. Nous allons donc maintenant détailler la construction de ces méta-modèles et les plans d'expériences associés.

a) Modèles linéaires – Plans factoriels

Les plans factoriels sont couramment utilisés pour construire des méta-modèles linéaires. Comme nous l'avons vu, la force des plans d'expériences réside dans le choix optimal des essais permettant d'estimer au mieux les coefficients du modèle. Pour les plans factoriels, le critère d'optimalité recherché est celui d'orthogonalité. Par la suite, nous traiterons trois types de modèles linéaires ainsi que les plans associés : les modèles linéaires complets, les modèles linéaires avec interactions d'ordre un et les modèles linéaires sans interactions.

Critère d'optimalité des plans factoriels : l'orthogonalité

Pour introduire le principe d'orthogonalité, on considère un plan d'expériences qui comporte n essais. La simulation de tous les essais permet d'obtenir n valeurs de la réponse. Cette réponse est ensuite approximée par un polynôme comportant q coefficients qui sont les inconnues du système suivant :

$$\mathbf{Y} = \mathbf{XA} + \boldsymbol{\varepsilon} \quad (4.6)$$

où $\mathbf{Y}$ est le vecteur des réponses, $\mathbf{X}$ la matrice du modèle qui dépend du plan d'expériences et du type de modèle, $\mathbf{A}$ le vecteur des coefficients polynomiaux et $\boldsymbol{\varepsilon}$ le vecteur des écarts modélisés par des variables aléatoires non-corrélées, de moyenne nulle et d'écart type σ . Il s'agit donc d'un système de n équations à q inconnues, ce qui impose au plan d'expériences d'avoir au moins autant d'essais qu'il y a de coefficients dans le modèle pour résoudre ce système par la méthode des moindres carrés. Cette méthode d'estimation consiste à déterminer les valeurs des coefficients qui minimisent la somme des carrés des écarts. Les valeurs des coefficients qui satisfont à ce critère sont données par l'estimateur des moindres carrés :

$$\mathbf{A} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y} \quad (4.7)$$

La matrice de variance-covariance associée à $\mathbf{A}$ est, quant à elle, définie par [104] :

$$\text{Var}(\mathbf{A}) = \sigma^2 (\mathbf{X}^T \mathbf{X})^{-1} \quad (4.8)$$

Cette matrice symétrique donne l'erreur sur les coefficients du modèle : les termes diagonaux correspondent à la variance des coefficients et les termes non-diagonaux à leur covariance. Il apparaît donc que l'erreur sur les coefficients dépend directement de la matrice du modèle X et donc des essais du plan d'expériences. C'est là le point fondamental de la théorie des plans d'expériences : identifier les essais à réaliser qui vont minimiser l'erreur sur les coefficients. Ainsi, on peut démontrer que si la matrice X est une matrice d'Hadamard, l'erreur sur les coefficients est minimale. Les matrices d'Hadamard, du nom du mathématicien français Jacques Hadamard (1865-1963), sont des matrices carrées orthogonales dont les termes sont -1 ou 1. L'expression (4.8) devient alors :

$$\text{Var}(A) = \sigma^2 n I_n \quad (4.9)$$

où I_n est la matrice identité et n le nombre d'essais du plan d'expériences. Ainsi, si X est une matrice d'Hadamard, la matrice de variance-covariance est une matrice diagonale ; les termes de covariance sont nuls. Les coefficients sont donc estimés de façon indépendante et l'on peut démontrer que leur variance est minimale [108].

Les modèles linéaires sont donc construits à partir de plans d'expériences qui satisfont à deux critères :

- le nombre d'essais est au moins égal au nombre de coefficients du modèle,
- leur structure est basée sur des matrices d'Hadamard dont l'orthogonalité permet d'estimer les coefficients du modèle avec la meilleure précision possible.

Modèle linéaire complet – Plans factoriels complets 2^k

Le modèle linéaire complet est un polynôme du premier degré. Pour trois facteurs, il a la forme suivante :

$$\hat{Y} = a_0 + b_1 X_1 + b_2 X_2 + b_3 X_3 + c_{12} X_1 X_2 + c_{13} X_1 X_3 + c_{23} X_2 X_3 + c_{123} X_1 X_2 X_3 \quad (4.10)$$

où a_0 est le terme constant, b_1 , b_2 et b_3 sont les coefficients des termes linéaires, c_{12} , c_{13} et c_{23} sont les coefficients des interactions d'ordre 1, enfin c_{123} est le coefficient de l'interaction d'ordre 2. S'il y a k facteurs à étudier, le modèle linéaire complet comporte donc 1 terme constant, k coefficients pour les termes linéaires et $C_k^2 + C_k^3 + \dots + C_k^k$ coefficients pour toutes les interactions. Ainsi, d'après la formule du binôme de Newton, le nombre total de coefficients à estimer est égal à 2^k . Les plans d'expériences requis pour construire un modèle linéaire complet comportent donc au minimum 2^k essais ; ces plans sont appelés plans factoriels complet 2^k . Chaque facteur pouvant prendre 2 niveaux, ils se construisent en étudiant toutes les combinaisons de niveaux possibles. Pour trois facteurs, un plan factoriel 2^3 comportera 8 essais ; sa matrice des essais est donnée par le Tableau

4.2. Avant d'estimer les coefficients du modèle par régression linéaire, il faut construire la matrice du modèle, notée X dans l'expression (4.7) : la colonne a_0 est associée au terme constant du modèle, les colonnes X_1, X_2, X_3 correspondent à la matrice des essais, tandis que les colonnes suivantes sont obtenues en multipliant ligne à ligne les colonnes de la matrices des essais pour représenter toutes les interactions du modèle. La matrice du modèle est donnée par le Tableau 4.3. On peut démontrer qu'il s'agit d'une matrice d'Hadamard. Les coefficients du modèle peuvent ensuite être calculés avec l'estimateur des moindres carrés (4.7) à partir de la matrice du modèle et des valeurs de la réponse aux points du plan d'expériences.

Tableau 4.2 : Matrice des essais d'un plan factoriel complet 2^3

	X_1	X_2	X_3
Essai n°1	-1	-1	-1
Essai n°2	+1	-1	-1
Essai n°3	-1	+1	-1
Essai n°4	+1	+1	-1
Essai n°5	-1	-1	+1
Essai n°6	+1	-1	+1
Essai n°7	-1	+1	+1
Essai n°8	+1	+1	+1

Tableau 4.3 : Matrice du modèle linéaire complet

	a_0	X_1	X_2	X_3	X_1X_2	X_1X_3	X_2X_3	$X_1X_2X_3$
Essai n°1	+1	-1	-1	-1	+1	+1	+1	-1
Essai n°2	+1	+1	-1	-1	-1	-1	-1	+1
Essai n°3	+1	-1	+1	-1	-1	+1	+1	+1
Essai n°4	+1	+1	+1	-1	+1	-1	-1	-1
Essai n°5	+1	-1	-1	+1	+1	-1	-1	+1
Essai n°6	+1	+1	-1	+1	-1	+1	+1	-1
Essai n°7	+1	-1	+1	+1	-1	-1	-1	-1
Essai n°8	+1	+1	+1	+1	+1	+1	+1	+1

En plus d'être orthogonaux, les plans factoriels complets présentent l'avantage d'être faciles à construire. Cependant le nombre d'essais augmente de façon exponentielle avec le nombre de facteurs (cf. Figure 4.5).

Modèle linéaire avec interactions d'ordre 1 – Plans factoriels fractionnaires de résolution V

Un plan factoriel complet permet d'estimer, à partir de 2^k essais, les effets des facteurs, mais également les interactions d'ordre 1, les interactions d'ordre 2 et toutes les interactions d'ordre supérieur. Or les interactions d'ordre 2, et plus, sont souvent négligeables ; les essais du plan d'expériences permettant d'estimer ces interactions pourraient donc être épargnés. Une solution consiste donc à mettre en place des plans fractionnaires : ce sont des plans factoriels dont le nombre d'essais a été divisé par 2, 4 ou 2^p par rapport à un plan complet.

L'idée de ces plans est d'utiliser la matrice du modèle d'un plan factoriel complet et de regrouper (on dit aussi « aliaser ») les facteurs supplémentaires à étudier avec des interactions dont on peut supposer que plus leur ordre est élevé plus elles sont négligeables. Ces groupes sont appelés « contrastes » ou « alias ». La notion de résolution permet de classer les plans fractionnaires en fonction des alias qui sont réalisés. Dans un plan de résolution III, l'effet d'un facteur est alié avec une ou plusieurs interactions d'ordre 1. Ce type de plan permet d'estimer les facteurs principaux, mais ne donne aucune information sur l'effet des interactions. Dans un plan de résolution IV, l'effet d'un facteur est alié avec une interaction d'ordre 2 et les interactions d'ordre 1 sont aliasées entre elles. Ce type de plan permet d'obtenir une meilleure estimation des facteurs, mais ne permet pas toujours de conclure sur les interactions d'ordre 1. Enfin, dans les plans de résolution V, l'effet d'un facteur est alié avec une interaction d'ordre 3, tandis que les interactions d'ordre 1 sont aliasées avec des interactions d'ordre 2. Les plans de résolution V permettent d'estimer tous les facteurs, ainsi que les interactions d'ordre 1. Plus la résolution d'un plan est élevée, meilleure est la quantification des facteurs et des interactions, mais plus d'essais sont nécessaires.

En supposant que les interactions d'ordre supérieur ou égal à 2 sont négligeables, le modèle linéaire avec les interactions d'ordre 1 est le suivant :

$$\hat{Y} = a_0 + \sum_{i=1}^k b_i X_i + \sum_{i=1}^{k-1} \sum_{q=i+1}^k c_{iq} X_i X_q \quad (4.11)$$

où k est le nombre de facteurs, a_0 le terme constant, b_i sont les coefficients des termes linéaires et c_{iq} les coefficients des interactions d'ordre 1. Ce modèle comporte $1 + k + C_k^2$ coefficients ; le plan fractionnaire associé doit donc comporter au moins autant d'essais. La construction de ces plans fait appel à la théorie des alias [103, 104] qui ne sera pas développée ici. De nombreux logiciels comme Matlab [109] ou Design Expert [110] permettent de bâtir de tels plans. La structure des plans fractionnaires reste basée sur des matrices d'Hadamard et satisfait donc au critère d'orthogonalité. Les plans factoriels fractionnaires sont notés plans $2^{k-p} = 2^k/2^p$ où 2^k correspond au plan complet pour

étudier k facteurs qui a été réduit de 2^p essais. Pour l'étude de cinq facteurs, un plan factoriel complet nécessiterait $2^5 = 32$ essais. Avec un plan factoriel fractionnaire 2^{5-1} de résolution V, les effets de tous les facteurs et des interactions d'ordre 1 peuvent être estimés avec 16 essais seulement ; le nombre d'essais a donc été divisé par deux par rapport à celui d'un plan factoriel complet. La matrice des essais du plan fractionnaire est donnée par le Tableau 4.4. La matrice du modèle nécessaire pour estimer les coefficients par régression linéaire (4.7) se bâtit de la même façon que pour les plans complets : reprendre la matrice des essais en lui ajoutant les colonnes correspondant aux interactions (cf. Tableau 4.5).

Tableau 4.4 : Matrice des essais d'un plan factoriel fractionnaire 2^{5-1}

	X_1	X_2	X_3	X_4	X_5
Essai n°1	-1	-1	-1	-1	+1
Essai n°2	+1	-1	-1	-1	-1
Essai n°3	-1	+1	-1	-1	-1
Essai n°4	+1	+1	-1	-1	+1
Essai n°5	-1	-1	+1	-1	-1
Essai n°6	+1	-1	+1	-1	+1
Essai n°7	-1	+1	+1	-1	+1
Essai n°8	+1	+1	+1	-1	-1
Essai n°9	-1	-1	-1	+1	-1
Essai n°10	+1	-1	-1	+1	+1
Essai n°11	-1	+1	-1	+1	+1
Essai n°12	+1	+1	-1	+1	-1
Essai n°13	-1	-1	+1	+1	+1
Essai n°14	+1	-1	+1	+1	-1
Essai n°15	-1	+1	+1	+1	-1
Essai n°16	+1	+1	+1	+1	+1

Tableau 4.5 : Matrice du modèle linéaire avec interactions d'ordre 1

	a_0	X_1	X_2	X_3	X_4	X_5	X_1X_2	X_1X_3	X_1X_4	X_1X_5	X_2X_3	X_2X_4	X_2X_5	X_3X_4	X_3X_5	X_4X_5
Essai n°1	+1	-1	-1	-1	-1	+1	+1	+1	+1	-1	+1	+1	-1	+1	-1	-1
Essai n°2	+1	+1	-1	-1	-1	-1	-1	-1	-1	-1	+1	+1	+1	+1	+1	+1
Essai n°3	+1	-1	+1	-1	-1	-1	-1	+1	+1	+1	-1	-1	-1	+1	+1	+1
Essai n°4	+1	+1	+1	-1	-1	+1	+1	-1	-1	+1	-1	-1	+1	+1	-1	-1
Essai n°5	+1	-1	-1	+1	-1	-1	+1	-1	+1	+1	-1	+1	+1	-1	-1	+1
Essai n°6	+1	+1	-1	+1	-1	+1	-1	+1	-1	+1	-1	+1	-1	-1	+1	-1
Essai n°7	+1	-1	+1	+1	-1	+1	-1	-1	+1	-1	+1	-1	+1	-1	+1	-1
Essai n°8	+1	+1	+1	+1	-1	-1	+1	+1	-1	-1	+1	-1	-1	-1	-1	+1
Essai n°9	+1	-1	-1	-1	+1	-1	+1	+1	-1	+1	+1	-1	+1	-1	+1	-1
Essai n°10	+1	+1	-1	-1	+1	+1	-1	-1	+1	+1	+1	-1	-1	-1	-1	+1
Essai n°11	+1	-1	+1	-1	+1	+1	-1	+1	-1	-1	-1	+1	+1	-1	-1	+1
Essai n°12	+1	+1	+1	-1	+1	-1	+1	-1	+1	-1	-1	+1	-1	-1	+1	-1
Essai n°13	+1	-1	-1	+1	+1	+1	+1	-1	-1	-1	-1	-1	-1	+1	+1	+1
Essai n°14	+1	+1	-1	+1	+1	-1	-1	+1	+1	-1	-1	-1	+1	+1	-1	-1
Essai n°15	+1	-1	+1	+1	+1	-1	-1	-1	-1	+1	+1	+1	-1	+1	-1	-1
Essai n°16	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1

Un modèle linéaire avec les interactions d'ordre 1 peut donc être construit à partir d'un plan factoriel fractionnaire de résolution V. Ces plans présentent l'avantage de nécessiter moins d'essais que les plans factoriels complets (cf. Figure 4.5), tout en conservant la propriété d'orthogonalité.

Modèle linéaire sans interactions – Plans de Plackett-Burman

Le dernier type de modèle linéaire présenté est le modèle purement additif qui ne comporte aucune interaction :

$$\hat{Y} = a_0 + \sum_{i=1}^k b_i X_i \quad (4.12)$$

où k est le nombre de facteurs, a_0 le terme constant et b_i sont les coefficients des termes linéaires. Le modèle linéaire sans interactions se compose donc de $1+k$ coefficients. Les plans mis en place pour construire ce type de modèle sont des plans de résolution III ; les plus connus d'entre eux sont les plans de Plackett-Burman [111]. Le principal avantage de ces plans est d'être très économiques en termes d'essais puisqu'ils ne requièrent au maximum que quatre essais de plus que le nombre de facteurs à étudier (leur nombre d'essais est un multiple de quatre). De plus, la matrice du modèle associée à ces plans est basée sur une matrice d'Hadamard ; Les plans de Plackett-Burman respectent donc le critère d'orthogonalité. La construction de ces plans est relativement simple : elle est basée sur la permutation circulaire d'un motif initial [104]. Pour étudier sept facteurs, la matrice des essais et la matrice du modèle sont données respectivement par le Tableau 4.6 et le Tableau 4.7.

Tableau 4.6 : Matrice des essais d'un plan de Plackett-Burman avec sept facteurs

	X_1	X_2	X_3	X_4	X_5	X_6	X_7
Essai n°1	+1	+1	+1	-1	+1	-1	-1
Essai n°2	-1	+1	+1	+1	-1	+1	-1
Essai n°3	-1	-1	+1	+1	+1	-1	+1
Essai n°4	+1	-1	-1	+1	+1	+1	-1
Essai n°5	-1	+1	-1	-1	+1	+1	+1
Essai n°6	+1	-1	+1	-1	-1	+1	+1
Essai n°7	+1	+1	-1	+1	-1	-1	+1
Essai n°8	-1	-1	-1	-1	-1	-1	-1

Tableau 4.7 : Matrice du modèle linéaire sans interactions avec sept facteurs

	a_0	X_1	X_2	X_3	X_4	X_5	X_6	X_7
Essai n°1	+1	+1	+1	+1	-1	+1	-1	-1
Essai n°2	+1	-1	+1	+1	+1	-1	+1	-1
Essai n°3	+1	-1	-1	+1	+1	+1	-1	+1
Essai n°4	+1	+1	-1	-1	+1	+1	+1	-1
Essai n°5	+1	-1	+1	-1	-1	+1	+1	+1
Essai n°6	+1	+1	-1	+1	-1	-1	+1	+1
Essai n°7	+1	+1	+1	-1	+1	-1	-1	+1
Essai n°8	+1	-1	-1	-1	-1	-1	-1	-1

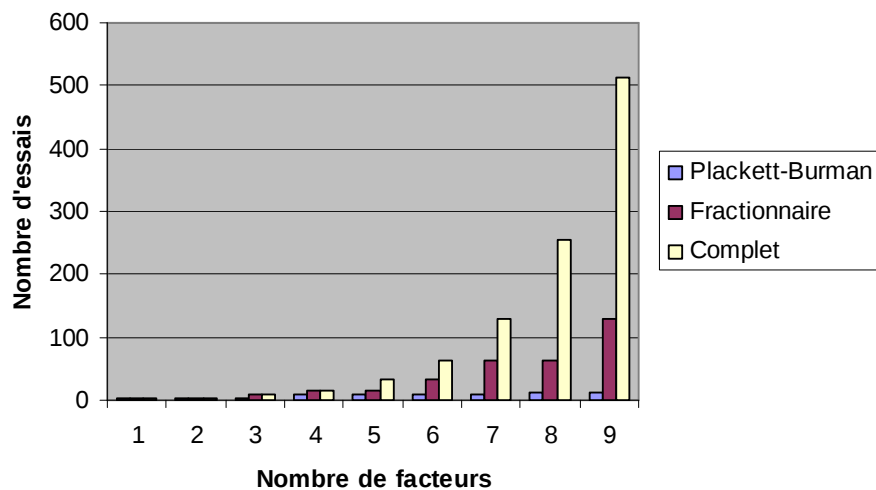
Le principal intérêt des plans de Plackett-Burman est de quantifier en peu d'essais l'influence de chaque facteur, l'influence d'un facteur étant donnée par la valeur de son coefficient dans le modèle. Ces plans sont généralement mis en place au début de la démarche de modélisation pour réduire le nombre de facteurs en supprimant les facteurs dont l'influence est négligeable sur la réponse. On parle alors de plans « de criblage » (« screening » en anglais).

Coût calculatoire des différents plans factoriels

Le coût calculatoire pour construire un méta-modèle dépend du coût pour réaliser la simulation des plans d'expériences et du coût de la régression linéaire pour calculer les coefficients du méta-modèle. Lorsque le nombre de facteurs est faible, le coût de la régression linéaire est généralement négligeable devant le coût des essais du plan d'expériences à simuler. Le Tableau 4.8 donne le nombre d'essais à réaliser pour les trois plans factoriels qui ont été présentés en fonction du nombre de facteurs étudiés. Le coût calculatoire des différents plans est également illustré par la Figure 4.5. D'un côté, les plans de Plackett-Burman nécessitent peu d'essais mais ne permettent d'analyser que les effets des facteurs principaux. De l'autre, les plans factoriels complets permettent de construire des modèles linéaires avec toutes les interactions au prix d'un très grand nombre de simulations. Il ressort donc que les plans factoriels fractionnaires de résolution V constituent un bon compromis : ils permettent de construire des modèles linéaires avec les interactions d'ordre 1 à partir de moins d'essais que les plans complets. Cependant, le modèle mathématique associé à ce type de plan est un polynôme du premier degré ; pour approximer des réponses plus complexes il faut avoir recours à un polynôme du second degré.

Tableau 4.8 : Coût calculatoire des plans factoriels en termes d'essais

Nombre de facteurs	Plans		
	Plackett-Burman	Fractionnaires	Complets
1	2	2	2
2	4	4	4
3	4	8	8
4	8	16	16
5	8	16	32
6	8	32	64
7	8	64	128
8	12	64	256
9	12	128	512

**Figure 4.5 : Nombre d'essais des plans de Plackett-Burman, des plans fractionnaires et des plans complets en fonction du nombre de facteurs étudiés.**

b) Modèle quadratique – Plans composites centrés

Si le modèle linéaire avec interactions n'est pas suffisant pour approximer la réponse du simulateur, un modèle quadratique peut se révéler plus adapté. Les plans composites centrés sont couramment utilisés pour construire de tels modèles à cause de leur approche séquentielle, comme nous allons l'expliquer.

Modèle quadratique avec interactions d'ordre 1

Un modèle quadratique avec les interactions d'ordre 1 peut s'exprimer sous la forme générale suivante :

$$\hat{Y} = a_0 + \sum_{i=1}^k b_i X_i + \sum_{i=1}^{k-1} \sum_{q=i+1}^k c_{iq} X_i X_q + \sum_{i=1}^k c_i X_i^2 \quad (4.13)$$

où k est le nombre de facteurs, a_0 le terme constant, b_i sont les coefficients des termes linéaires, c_{iq} les coefficients des interactions d'ordre 1 et c_i les coefficients des termes quadratiques. Ce modèle comporte donc $1 + 2k + C_k^2$ coefficients à estimer.

Plans composites centrés

Pour estimer la courbure du modèle quadratique, les essais à mener doivent être répartis sur au moins trois niveaux. Parmi les plans de trois niveaux ou plus, les plans composites centrés [112] sont souvent utilisés à cause de leur construction séquentielle. Ces plans se composent :

- d'un plan factoriel complet ou fractionnaire (points rouges sur la Figure 4.6) pour estimer les coefficients des termes linéaires et des interactions,
- de points en étoile augmentés d'un point au centre (points bleus sur la Figure 4.6) pour estimer les coefficients des termes quadratiques. Il y a deux points en étoile par facteur ; ces points ont pour valeurs $-\alpha$ et $+\alpha$.

Le nombre total d'essais à réaliser est donc égal à $N_f + 1 + 2k$ où N_f est le nombre d'essais du plan factoriel. Le plan composite centré implémenté pour l'étude de deux facteurs est illustré par la Figure 4.6. Sa matrice des essais est représentée par le Tableau 4.9 et sa matrice du modèle par le Tableau 4.10.

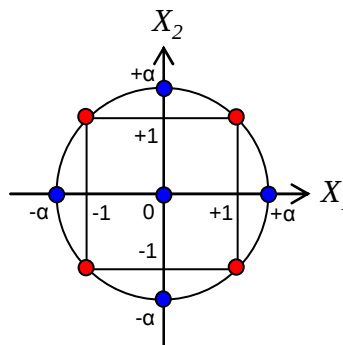


Figure 4.6 : Plan composite centré avec deux facteurs.

Tableau 4.9 : Matrice des essais d'un plan composite centré à deux facteurs

	X_1	X_2
Essai n°1	-1	-1
Essai n°2	+1	-1
Essai n°3	-1	+1
Essai n°4	+1	+1
Essai n°5	0	0
Essai n°6	$-\alpha$	0
Essai n°7	$+\alpha$	0
Essai n°8	0	$-\alpha$
Essai n°9	0	$+\alpha$

Tableau 4.10 : Matrice du modèle quadratique avec deux facteurs

	a_0	X_1	X_2	X_1X_2	X_1^2	X_2^2
Essai n°1	+1	-1	-1	+1	+1	+1
Essai n°2	+1	+1	-1	-1	+1	+1
Essai n°3	+1	-1	+1	-1	+1	+1
Essai n°4	+1	+1	+1	+1	+1	+1
Essai n°5	+1	0	0	0	0	0
Essai n°6	+1	$-\alpha$	0	0	$+\alpha^2$	0
Essai n°7	+1	$+\alpha$	0	0	$+\alpha^2$	0
Essai n°8	+1	0	$-\alpha$	0	0	$+\alpha^2$
Essai n°9	+1	0	$+\alpha$	0	0	$+\alpha^2$

Critère d'optimalité

Dans le cas des plans composites, la matrice du modèle n'est plus une matrice d'Hadamard, le critère d'orthogonalité n'est donc pas vérifié (sauf dans certains cas très particuliers). Un autre critère d'optimalité est donc recherché pour ces plans : il s'agit en général de l'isovariance par rotation. Ce critère consiste à assurer que la variance des réponses prédites par le modèle est identique pour les points situés à la même distance du centre du domaine d'étude. L'isovariance est atteinte lorsque le paramètre α des points en étoile vérifie [104] :

$$\alpha = (N_f)^{\frac{1}{4}} \quad (4.14)$$

où N_f est le nombre d'essais du plan factoriel. Les facteurs d'un tel plan composite sont donc répartis sur cinq niveaux, ce qui les rend plus complexes à mettre en œuvre. Des plans plus simples peuvent être obtenus en fixant la valeur du paramètre α à 1 ; il n'y a alors que trois niveaux par facteurs, mais la propriété d'isovariance n'est plus vérifiée (cf. Figure 4.7).

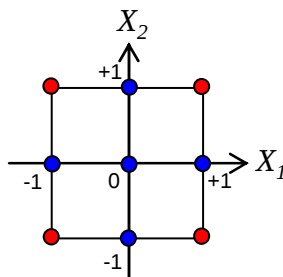


Figure 4.7 : Plan composite centré avec $\alpha = 1$

Construction séquentielle des plans composites centrés

Le principal intérêt des plans composites centrés réside dans leur construction progressive. En effet, un plan factoriel complet ou fractionnaire peut être tout d'abord mis en place pour construire un modèle linéaire. Si ce modèle linéaire n'est pas assez précis pour prédire la réponse du simulateur, les essais du plan factoriel peuvent être réutilisés dans un plan composite pour construire un modèle quadratique ; seuls les points en étoile et le point au centre doivent alors être simulés.

c) Validation des modèles

La précision des modèles linéaires ou quadratiques est mesurée à partir des grandeurs présentées plus haut ($RMSE$, MAX et R^2). Le calcul de ces grandeurs nécessite la simulation d'essais supplémentaires en plus des essais simulés pour construire le modèle. Plusieurs choix sont possibles pour l'emplacement des essais supplémentaires. Une première possibilité peut consister à les prendre au hasard dans le domaine d'étude. Une seconde possibilité, plus efficace, consiste à placer ces points supplémentaires de telle manière qu'ils permettent de mesurer la précision du modèle loin des points ayant servi à sa construction. Dans le cas d'un modèle linéaire, les points supplémentaires peuvent correspondre aux points en étoile plus le point au centre (voir Figure 4.8(a)). Ainsi, en cas de non-validité du modèle linéaire, la construction d'un modèle quadratique par la suite ne demandera pas d'autres simulations. Dans le cas d'un modèle quadratique, les points supplémentaires peuvent se répartir à équidistance des points intervenant dans la construction du modèle comme indiqué sur la Figure 4.8(b). En pratique, dans les circuits étudiés, un nombre d'essais supplémentaires égal à environ 10% du nombre d'essais du plan d'expériences est suffisant pour estimer correctement les différents critères de validation.

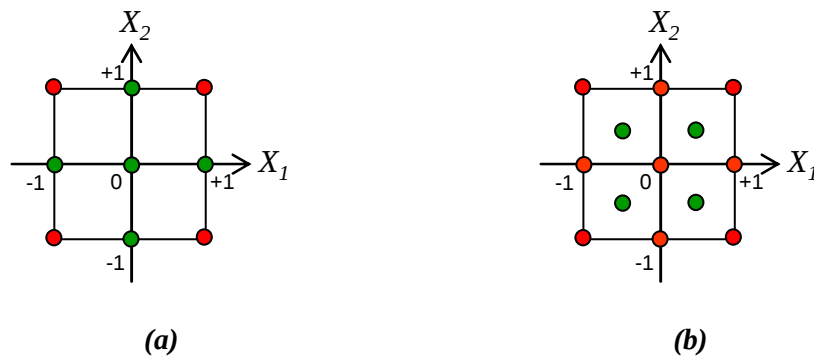


Figure 4.8 : Emplacement des points supplémentaires (en vert) pour tester un modèle linéaire (a) ou quadratique (b)

d) Comparaison des modèles polynomiaux

Les caractéristiques des différents modèles que nous avons abordés sont résumées dans le Tableau 4.11. En termes de modélisation, les modèles les plus intéressants sont le modèle linéaire avec les interactions d'ordre 1 et le modèle quadratique. En effet, la prise en compte des interactions d'ordre 1 apporte plus de flexibilité pour approximer des réponses non-linéaires. De plus, en considérant comme négligeables les interactions d'ordre supérieur ou égal à 2, le modèle linéaire avec interactions d'ordre 1 possède moins de coefficients à estimer que le modèle linéaire complet ; le plan d'expériences mis en place pour le construire comporte donc moins d'essais. Quant au modèle quadratique, sa construction à partir d'un plan composite centré permet de réutiliser les simulations réalisées pour bâtir un modèle linéaire, diminuant ainsi le coût calculatoire. Il faut cependant rappeler que les modèles polynomiaux restent intéressants tant que le nombre de facteurs n'excède pas la dizaine. En effet, au-delà de 21 facteurs il faut plus de 1000 simulations pour construire un modèle quadratique ; d'autres types de méta-modèles moins coûteux en calcul sont alors plus adaptés (krigeage, fonctions à base radiale).

Tableau 4.11 : Comparaison des caractéristiques des modèles linéaires et quadratiques

	Modèle linéaire sans interactions	Modèle linéaire avec interactions d'ordre 1	Modèle linéaire complet	Modèle quadratique avec interactions d'ordre 1
Nombre de facteurs	k	k	k	k
Nombre de coefficients	$1 + k$	$1 + k + C_k^2$	2^k	$1 + 2k + C_k^2$
Plan d'expériences	Plackett-Burman	Factoriel fractionnaire de résolution V	Factoriel complet	Composite centré
Nombre d'essais	$(1 + \text{Ent}(k/4)) \times 4$	2^{k-p}	2^k	$2^{k-p} + 1 + 2k$
Propriétés	Recherche des facteurs influents en peu d'essais	Bon compromis entre la précision du modèle et le nombre d'essais	Le nombre d'essais est prohibitif	Construction séquentielle

* $\text{Ent}(\)$ représente la fonction partie entière

4.2.4 Approximation des performances par des polynômes

Dans ce travail de thèse, les méta-modèles utilisés pour approximer les performances des circuits analogiques seront des polynômes d'ordre un ou deux :

$$\hat{Y} = \hat{f}(C, \Delta T, E) \quad (4.15)$$

où $\hat{Y}$ est l'approximation d'une performance, $\hat{f}$ correspond à un méta-modèle polynomial d'ordre un ou deux, tandis que C , ΔT et E correspondent respectivement au vecteur des paramètres de conception, au vecteur des variations des paramètres technologiques et au vecteur des paramètres environnementaux. Cette approche de modélisation des performances par des polynômes est identique à celle mise en place dans [28, 113].

Comme nous l'avons vu au §4.2.2b), les polynômes d'ordre un ou deux sont bien adaptés pour réaliser des approximations locales : sur des domaines d'étude trop larges, leurs prédictions peuvent être de moins bonne qualité en cas de non-linéarité de la réponse. Il se peut donc que le domaine de conception D et la région de tolérance des paramètres environnementaux RT_E doivent être réduits pour construire une approximation polynomiale valide. En ce qui concerne la région de tolérance des variations des paramètres technologiques $RT_{\Delta T}$ (définie par les intervalles $[-3\sigma, +3\sigma]$ de chaque

variation), elle ne peut en aucun cas être rétrécie afin de toujours tenir compte de toutes les variations possibles. Sur les circuits testés, les modèles quadratiques se sont avérés suffisants pour approximer les performances sur la région de tolérance des variations technologiques $RT_{\Delta T}$.

Une autre contrainte des polynômes est le nombre maximum de paramètres qui peuvent être considérés : au plus 20 (cf. §4.2.3), alors que les circuits analogiques peuvent dépendre de plusieurs centaines de paramètres. Cependant, rarement tous les paramètres ont une influence significative sur chaque performance. En effet, pour une performance donnée, seule une poignée d'entre eux ont un impact non-négligeable. Ces paramètres influents peuvent être facilement identifiés grâce à un plan de criblage peu coûteux en simulations de type Plackett-Burman et une analyse de sensibilité. Cette approche permet alors de réduire le nombre de paramètres à considérer dans la construction des méta-modèles polynomiaux des performances.

4.3 Estimation des bornes de variation des performances

Dans la partie précédente, nous avons vu comment approximer les performances d'un circuit analogique par des modèles linéaires ou quadratiques. Dans cette partie, nous nous intéressons à l'impact des variations paramétriques ΔT sur les performances, pour des valeurs de paramètres de conception C et environnementaux E données. Les méta-modèles des performances sont donc de la forme : $\hat{Y} = \hat{f}_{C_0, T_0}(\Delta T)$. De plus, nous considérons que les variations technologiques ΔT sont distribuées selon des lois normales de moyenne nulle et de matrice de covariance Σ . A présent, nous allons expliquer les méthodes mises en place pour estimer les bornes de variation des performances (valeurs « pire-cas »), d'une part pour des modèles linéaires sans interactions, d'autre part pour des modèles linéaires avec interactions ou des modèles quadratiques.

4.3.1 Bornes de variation des performances – Valeurs pire-cas

La méthode proposée dans cette thèse pour caractériser la variabilité d'une performance consiste à définir son domaine de variation. En effet, en modélisant une performance par une variable aléatoire Y de densité de probabilité $f_Y(y)$, on peut alors calculer la probabilité p que Y appartienne à l'intervalle $[a, b[$:

$$p = P(a \leq Y < b) = \int_a^b f_Y(y) dy \quad (4.16)$$

En se basant sur cette définition, on peut considérer le domaine de variation d'une performance comme un intervalle $[a, b]$ qui a une probabilité p de contenir les valeurs de la performance. Pour l'analyse de variabilité des circuits, toutes les valeurs possibles d'une performance, en particulier les valeurs extrêmes, doivent être prises en compte ; l'intervalle considéré doit donc englober la quasi-totalité des valeurs. Ainsi en prenant les 1^{er} et 99^{ème} centiles de la distribution de la performance comme bornes du domaine de variation, la probabilité de contenir les valeurs de la performance sera égale à 0.98 (cf. Figure 4.9). Ces bornes sont appelées valeurs « pire-cas » de la performance : la borne inférieure est notée $\hat{Y}_{\min}$ et la borne supérieure $\hat{Y}_{\max}$. L'objectif de la méthode proposée pour analyser la variabilité des circuits analogiques est d'estimer directement ces valeurs pire-cas.

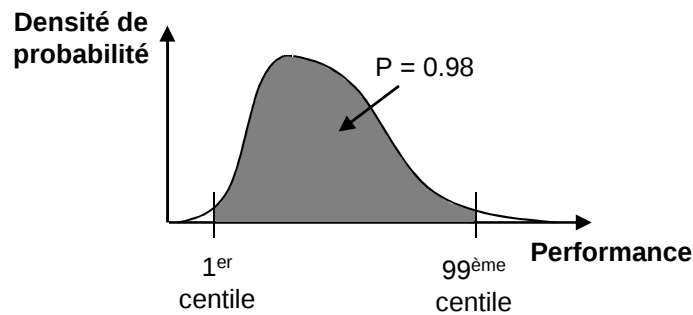


Figure 4.9 : Bornes de variation d'une performance

4.3.2 Modèle linéaire sans interactions

Nous allons tout d'abord nous intéresser à l'estimation des valeurs pire-cas dans le cas d'un modèle linéaire sans interactions. Soit une performance Y approximée par un modèle linéaire sans interactions $\hat{Y}$:

$$\hat{Y} = a_0 + \sum_{i=1}^k b_i X_i \quad (4.17)$$

où les X_i sont des variables aléatoires (qui représentent les variations des paramètres technologiques). Si ces variables aléatoires sont distribuées selon des lois normales de moyenne nulle et de matrice de covariance Σ , alors l'approximation $\hat{Y}$ est également une loi normale. En écrivant l'expression (4.17) sous forme matricielle :

$$\hat{Y} = a_0 + \mathbf{B}^T \mathbf{X} \quad (4.18)$$

où $\mathbf{B}$ est le vecteur des coefficients linéaires du modèle et $\mathbf{X}$ le vecteur des variables aléatoires, la moyenne $\mu_{\hat{Y}}$ et la variance $\sigma_{\hat{Y}}^2$ de $\hat{Y}$ sont alors données par les expressions (4.19) et (4.20).

$$\mu_{\hat{Y}} = a_0 \quad (4.19)$$

$$\sigma_{\hat{Y}}^2 = \mathbf{B}^T \boldsymbol{\Sigma} \mathbf{B} \quad (4.20)$$

Les 1^{er} et 99^{ème} centiles de $\hat{Y}$ peuvent alors se calculer à partir des expressions (4.21) et (4.22) :

$$\hat{Y}_{\min} = \hat{Y}_{0.01} = \mu_{\hat{Y}} + Z_{0.01} \sigma_{\hat{Y}} \quad (4.21)$$

$$\hat{Y}_{\max} = \hat{Y}_{0.99} = \mu_{\hat{Y}} + Z_{0.99} \sigma_{\hat{Y}} \quad (4.22)$$

où $Z_{0.01}$ et $Z_{0.99}$ sont respectivement les 1^{er} et 99^{ème} centiles d'une loi normale centrée réduite $N(0,1)$. L'estimation des valeurs pire-cas pour un modèle linéaire sans interactions est donc relativement simple : seules la moyenne et la variance sont nécessaires pour caractériser une loi normale et ainsi calculer les centiles correspondant aux valeurs pire-cas.

4.3.3 Modèle linéaire avec interactions et modèle quadratique – Développement limité de Cornish-Fisher

Dans le cas d'un modèle linéaire avec interactions ou d'un modèle quadratique, il n'y a plus de linéarité entre l'approximation de la performance $\hat{Y}$ et les paramètres technologiques. La distribution de $\hat{Y}$ ne suit donc pas une loi normale. La moyenne et la variance seules ne permettent alors pas d'estimer les valeurs pire-cas. Cependant, nous allons montrer dans cette partie qu'il est possible d'estimer ces valeurs extrêmes à partir des cumulants de $\hat{Y}$ et du développement limité de Cornish-Fisher.

a) Forme quadratique – loi du χ^2

Le calcul des cumulants d'une forme quadratique n'est pas immédiat. Il faut au préalable appliquer une transformation afin d'exprimer le modèle quadratique sous la forme d'une somme de variables aléatoires indépendantes qui suivent des lois du χ^2 [114, 115].

Soit une performance Y approximée par un modèle linéaire avec les interactions d'ordre 1 (4.23) ou par un modèle quadratique (4.24) :

$$\hat{Y} = a_0 + \sum_{i=1}^k b_i X_i + \sum_{i=1}^{k-1} \sum_{q=i+1}^k c_{iq} X_i X_q \quad (4.23)$$

$$\hat{Y} = a_0 + \sum_{i=1}^k b_i X_i + \sum_{i=1}^{k-1} \sum_{q=i+1}^k c_{iq} X_i X_q + \sum_{i=1}^k c_i X_i^2 \quad (4.24)$$

où les X_i sont des variables aléatoires qui suivent des lois normales de moyenne nulle et de matrice de covariance Σ . Les expressions (4.23) et (4.24) peuvent s'écrire plus généralement sous la forme matricielle suivante :

$$\hat{Y} = a_0 + \mathbf{B}^T \mathbf{X} + \frac{1}{2} \mathbf{X}^T \mathbf{C} \mathbf{X} \quad (4.25)$$

où $\mathbf{B}$ est le vecteur des coefficients linéaires, $\mathbf{C}$ une matrice carrée représentant les coefficients quadratiques (termes diagonaux) ainsi que les interactions (termes non-diagonaux) et $\mathbf{X}$ le vecteur des variables aléatoires normales.

Soit $\mathbf{Z}$ une matrice qui vérifie :

$$\mathbf{Z}\mathbf{Z}^T = \Sigma \quad (4.26)$$

Cette matrice $\mathbf{Z}$ peut être obtenue en appliquant la factorisation de Cholesky. On diagonalise la matrice $\mathbf{Z}^T \mathbf{C} \mathbf{Z}$ en déterminant une matrice de passage orthogonale $\mathbf{U}$ dont les vecteurs colonnes sont les vecteurs propres de $\mathbf{Z}^T \mathbf{C} \mathbf{Z}$. On a alors la relation suivante :

$$\mathbf{U}^T \mathbf{Z}^T \mathbf{C} \mathbf{Z} \mathbf{U} = \dot{\mathbf{C}} \quad (4.27)$$

où $\dot{\mathbf{C}}$ est une matrice diagonale. On définit alors le changement de variables suivant [114] :

$$\dot{\mathbf{X}} = \mathbf{U}^T \mathbf{Z}^{-1} \mathbf{X} \quad (4.28)$$

où $\dot{\mathbf{X}}$ est un vecteur de variables aléatoires de moyenne nulle :

$$\begin{aligned} E(\dot{\mathbf{X}}) &= E(\mathbf{U}^T \mathbf{Z}^{-1} \mathbf{X}) \\ E(\dot{\mathbf{X}}) &= \mathbf{U}^T \mathbf{Z}^{-1} E(\mathbf{X}) \\ E(\dot{\mathbf{X}}) &= 0 \end{aligned} \quad (4.29)$$

et de matrice de covariance $\dot{\Sigma}$:

$$\begin{aligned} \dot{\Sigma} &= E\left((\dot{\mathbf{X}} - E(\dot{\mathbf{X}}))(\dot{\mathbf{X}} - E(\dot{\mathbf{X}}))^T\right) \\ \dot{\Sigma} &= \mathbf{U}^T \mathbf{Z}^{-1} E\left((\mathbf{X} - E(\mathbf{X}))(\mathbf{X} - E(\mathbf{X}))^T\right) (\mathbf{U}^T \mathbf{Z}^{-1})^T \\ \dot{\Sigma} &= \mathbf{U}^T \mathbf{Z}^{-1} \Sigma (\mathbf{U}^T \mathbf{Z}^{-1})^T \\ \dot{\Sigma} &= \mathbf{U}^T \mathbf{Z}^{-1} \Sigma (\mathbf{Z}^T)^{-1} \mathbf{U} \\ \dot{\Sigma} &= \mathbf{U}^T \mathbf{Z}^T (\mathbf{Z}^T)^{-1} \mathbf{U} \\ \dot{\Sigma} &= \mathbf{I} \end{aligned} \quad (4.30)$$

Ainsi la nouvelle variable $\dot{\mathbf{X}}$ est un vecteur de variables aléatoires indépendantes qui suivent des lois normales centrées réduites $N(0,1)$. En appliquant le changement de variables dans l'expression (4.25), on obtient :

$$\begin{aligned}\hat{Y} &= a_0 + \mathbf{B}^T \mathbf{Z} \mathbf{U} \dot{\mathbf{X}} + \frac{1}{2} (\mathbf{Z} \mathbf{U} \dot{\mathbf{X}})^T \mathbf{C} \mathbf{Z} \mathbf{U} \dot{\mathbf{X}} \\ \hat{Y} &= a_0 + (\dot{\mathbf{X}}^T \mathbf{U}^T \mathbf{Z}^T \mathbf{B})^T + \frac{1}{2} \dot{\mathbf{X}}^T \mathbf{U}^T \mathbf{Z}^T \mathbf{C} \mathbf{Z} \mathbf{U} \dot{\mathbf{X}} \\ \hat{Y} &= a_0 + \dot{\mathbf{B}}^T \dot{\mathbf{X}} + \frac{1}{2} \dot{\mathbf{X}}^T \dot{\mathbf{C}} \dot{\mathbf{X}}\end{aligned}\tag{4.31}$$

avec $\dot{\mathbf{B}} = \mathbf{U}^T \mathbf{Z}^T \mathbf{B}$. Avec ce changement de variables, la forme quadratique $\hat{Y}$ ne contient plus de termes d'interactions puisque la matrice $\dot{\mathbf{C}}$ est diagonale. On peut donc écrire $\hat{Y}$ sous la forme :

$$\begin{aligned}\hat{Y} &= a_0 + \sum_{i=1}^k \left(\dot{b}_i \dot{X}_i + \frac{1}{2} \dot{c}_i \dot{X}_i^2 \right) \\ \hat{Y} &= a_0 + \sum_{i=1}^k \left[\frac{1}{2} \dot{c}_i \left(\frac{\dot{b}_i}{\dot{c}_i} + \dot{X}_i \right)^2 - \frac{\dot{b}_i^2}{2\dot{c}_i} \right]\end{aligned}\tag{4.32}$$

où les $\dot{b}_i$ sont les composantes du vecteur $\dot{\mathbf{B}}$ et les $\dot{c}_i$ les termes diagonaux de la matrice $\dot{\mathbf{C}}$. Puisque $\dot{X}_i$ est une variable normale centrée réduite, chaque terme $\left(\dot{b}_i/\dot{c}_i + \dot{X}_i \right)^2$ suit une loi du Khi-2 non-centrée $\chi^2 \left(1, \left(\dot{b}_i/\dot{c}_i \right)^2 \right)$ (cf. Annexe A). Grâce à cette loi, dont on connaît la fonction caractéristique, on va pouvoir calculer les cumulants d'une forme quadratique.

b) Cumulants d'une forme quadratique

Les cumulants κ_n d'une variable aléatoire X sont obtenus à partir du logarithme de sa fonction caractéristique $\varphi_X(t)$ (voir Annexe A) :

$$\ln(\varphi_X(t)) = \ln(E(e^{jtX}))\tag{4.33}$$

Ils correspondent aux coefficients du développement en série de MacLaurin de (4.33) :

$$\ln(\varphi_X(t)) = \sum_{n=1}^{\infty} \kappa_n \frac{(jt)^n}{n!}\tag{4.34}$$

Grâce au changement de variables (4.28), la forme quadratique $\hat{Y}$ (4.32) peut désormais s'exprimer à partir des variables aléatoires $\left(\dot{b}_i/\dot{c}_i + \dot{X}_i\right)^2$ qui suivent une loi du Khi-2 non-centrée $\chi^2\left(1, \left(\dot{b}_i/\dot{c}_i\right)^2\right)$. La fonction caractéristique de cette loi s'écrit ainsi (cf. Annexe A) :

$$\varphi(t) = \frac{\exp\left(\frac{jt}{1-2jt}\left(\frac{\dot{b}_i}{\dot{c}_i}\right)^2\right)}{\sqrt{1-2jt}} \quad (4.35)$$

Avec (4.35), on peut donc calculer la fonction caractéristique de chaque terme $\dot{b}_i \dot{X}_i + \frac{1}{2} \dot{c}_i \dot{X}_i^2$ de la forme quadratique (4.32) :

$$E\left(e^{jt\left(\dot{b}_i \dot{X}_i + \frac{1}{2} \dot{c}_i \dot{X}_i^2\right)}\right) = E\left(e^{jt\left(\frac{1}{2} \dot{c}_i \left(\frac{\dot{b}_i}{\dot{c}_i} + \dot{X}_i\right)^2 - \frac{\dot{b}_i^2}{2\dot{c}_i}\right)}\right) = \frac{1}{\sqrt{1-\dot{c}_i jt}} \exp\left(\frac{\dot{b}_i^2 (jt)^2}{2(1-\dot{c}_i jt)}\right) \quad (4.36)$$

d'où l'on tire la fonction caractéristique de la forme quadratique $\hat{Y}$:

$$\varphi_{\hat{Y}}(t) = E(e^{jt\hat{Y}}) = e^{jta_0} \prod_{i=1}^k \left(\frac{1}{\sqrt{1-\dot{c}_i jt}} \exp\left(\frac{\dot{b}_i^2 (jt)^2}{2(1-\dot{c}_i jt)}\right) \right) \quad (4.37)$$

puis son logarithme :

$$\ln(\varphi_{\hat{Y}}(t)) = a_0 jt + \sum_{i=1}^k \left(\frac{\dot{b}_i^2 (jt)^2}{2(1-\dot{c}_i jt)} - \frac{1}{2} \ln(1-\dot{c}_i jt) \right) \quad (4.38)$$

En introduisant les développements de MacLaurin de $(1-x)^{-1}$ et $\ln(1-x)$ dans (4.38), on obtient finalement les cumulants κ_n de $\hat{Y}$ [114]. Pour $n = 1$:

$$\begin{aligned} \kappa_1 &= a_0 + \frac{1}{2} \sum_{i=1}^k \dot{c}_i \\ \kappa_1 &= a_0 + \frac{1}{2} \text{tr}(\mathbf{C}\boldsymbol{\Sigma}) \end{aligned} \quad (4.39)$$

où $\text{tr}()$ est la trace d'une matrice. Pour $n \geq 2$:

$$\kappa_n = \frac{1}{2} \sum_{i=1}^k \left[(n-1)! \dot{c}_i^n + n! b_i^2 \dot{c}_i^{n-2} \right] \quad (4.40)$$

$$\kappa_n = \frac{1}{2} (n-1)! \text{tr}((C\Sigma)^n) + \frac{1}{2} n! \mathbf{B}^T \Sigma (C\Sigma)^{n-2} \mathbf{B}$$

Ainsi tous les cumulants d'une forme quadratique peuvent se calculer à partir de son terme constant a_0 , de ses coefficients $\mathbf{B}$ et $\mathbf{C}$ dans l'expression (4.25), et de la matrice de covariance Σ des variables aléatoires normales. Le calcul de ces cumulants va nous permettre à présent d'estimer les bornes de variation des performances en utilisant le développement limité de Cornish-Fisher.

c) Développement limité de Cornish-Fisher

L'approximation de Cornish-Fisher est un développement limité qui permet d'approcher le centile d'une distribution à partir de ses cumulants et du centile d'une loi normale. Historiquement, il fut développé en 1937 par E. A. Cornish (1909-1973) et R. A. Fisher (1890-1962) [116]. De nos jours, il est utilisé en mathématiques financières pour évaluer les risques dans les portefeuilles d'actions en présence de distributions non-gaussiennes [117]. Le développement de Cornish-Fisher se révèle en effet particulièrement bien adapté pour étudier les queues de telles distributions.

Soit Y une variable aléatoire normalisée (moyenne nulle, variance égale à 1) et Y_α son centile que l'on cherche à approximer et qui est associé à la probabilité α . Soit Z une variable aléatoire normale centrée réduite $N(0,1)$ et Z_α son centile associé à cette même probabilité α (cf. Figure 4.10).

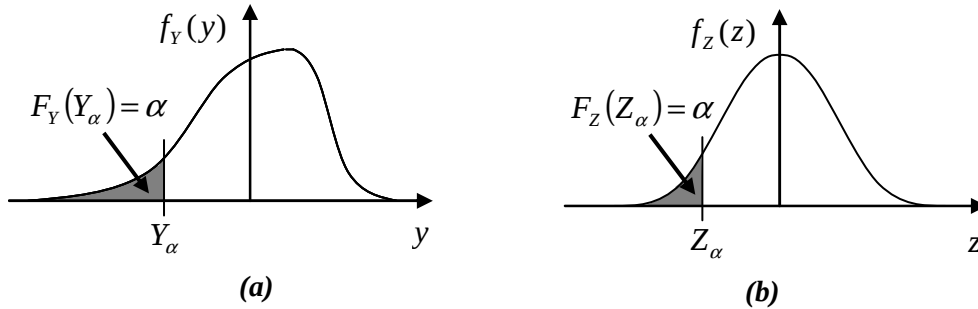


Figure 4.10 : Densité de probabilité de Y (a) et d'une loi normale centrée réduite (b)

L'approximation de Cornish-Fisher vise à déterminer la valeur de Y_α qui vérifie [118] :

$$\int_{-\infty}^{Y_\alpha} f_Y(y) dy = \int_{-\infty}^{Z_\alpha} f_Z(z) dz = \alpha \quad (4.41)$$

$$\Leftrightarrow F_Y(Y_\alpha) = F_Z(Z_\alpha) = \alpha$$

où f_Y et F_Y sont respectivement la densité de probabilité et la fonction de répartition de Y , tandis que f_Z et F_Z celles de Z . Le développement de Cornish-Fisher s'obtient à partir de (4.41) en approximant tout d'abord la fonction de répartition F_Y par le développement limité de Gram-Charlier

[118], puis en l'inversant pour déterminer $F_Y^{-1}(\alpha) = Y_\alpha$. Le développement limité de Cornish-Fisher avec les cinq premiers cumulants κ_n s'écrit alors [118] :

$$\begin{aligned} Y_\alpha = Z_\alpha &+ \frac{1}{6}(Z_\alpha^2 - 1)\kappa_3 + \frac{1}{24}(Z_\alpha^3 - 3Z_\alpha)\kappa_4 - \frac{1}{36}(2Z_\alpha^3 - 5Z_\alpha)\kappa_3^2 \\ &+ \frac{1}{120}(Z_\alpha^4 - 6Z_\alpha^2 + 3)\kappa_5 - \frac{1}{24}(Z_\alpha^4 - 5Z_\alpha^2 + 2)\kappa_3\kappa_4 \\ &+ \frac{1}{324}(12Z_\alpha^4 - 53Z_\alpha^2 + 17)\kappa_3^3 + \dots \end{aligned} \quad (4.42)$$

Le développement limité de Cornish-Fisher permet donc d'approcher le centile Y_α de Y à partir de ses cumulants κ_n et du centile Z_α de la loi normale centrée réduite $N(0,1)$.

Dans la suite cette thèse, nous utiliserons le développement de Cornish-Fisher pour estimer les valeurs pire-cas d'une performance approximée par une forme quadratique $\hat{Y}$ qui n'est en général pas normalisée. Or le développement limité de Cornish-Fisher ne s'applique qu'à des variables normalisées. On normalise donc $\hat{Y}$ en effectuant le changement de variables suivant :

$$\hat{Y}^* = \frac{\hat{Y} - \mu_{\hat{Y}}}{\sigma_{\hat{Y}}} \quad (4.43)$$

où $\mu_{\hat{Y}}$ est la moyenne de $\hat{Y}$ et $\sigma_{\hat{Y}}$ son écart type. Les cumulants normalisés $\kappa_{n,\hat{Y}}^*$ de $\hat{Y}^*$ se calculent avec les cumulants $\kappa_{n,\hat{Y}}$ de $\hat{Y}$:

$$\kappa_{n,\hat{Y}}^* = \frac{\kappa_{n,\hat{Y}}}{\sigma_{\hat{Y}}^n} \quad (4.44)$$

Les 1^{er} et 99^{ème} centiles de $\hat{Y}$ peuvent alors s'exprimer à partir des centiles normalisés $\hat{Y}_{0.01}^*$ et $\hat{Y}_{0.99}^*$ de $\hat{Y}^*$ obtenus avec l'approximation de Cornish-Fisher (4.42), en appliquant le changement de variables inverse de (4.43) :

$$\hat{Y}_{\min} = \hat{Y}_{0.01} = \mu_{\hat{Y}} + \hat{Y}_{0.01}^* \sigma_{\hat{Y}} \quad (4.45)$$

$$\hat{Y}_{\max} = \hat{Y}_{0.99} = \mu_{\hat{Y}} + \hat{Y}_{0.99}^* \sigma_{\hat{Y}} \quad (4.46)$$

Pour résumer, l'estimation des valeurs pire-cas pour un modèle linéaire avec interactions (4.23) ou quadratique (4.24) s'effectue donc en quatre étapes :

- les cumulants $\kappa_{n,\hat{Y}}$ des modèles sont d'abord calculés avec les expressions (4.39) et (4.40),
- les cumulants normalisés $\kappa_{n,\hat{Y}}^*$ sont ensuite déterminés à partir de l'expression (4.44),
- les 1^{er} et 99^{ème} centiles normalisés sont alors obtenus avec l'approximation de Cornish-Fisher (4.42),
- enfin, les valeurs pire-cas des modèle sont tirées des expressions (4.45) et (4.46) à partir des 1^{er} et 99^{ème} centiles normalisés. On peut remarquer que dans le cas particulier où $\hat{Y}^*$ suit une loi normale centrée réduite, les cumulants d'ordre supérieur à 3 sont nuls, les expressions (4.45) et (4.46) sont alors identiques aux expressions (4.21) et (4.22) obtenues pour le modèle linéaire.

Grâce au développement limité de Cornish-Fisher, n'importe quel centile d'une variable aléatoire dont la distribution est inconnue peut être approximé à partir de ses cumulants et du centile correspondant d'une loi normale. Néanmoins, les travaux réalisés dans [114] ont montré que la qualité de l'estimation de Cornish-Fisher se dégradait pour les centiles très extrêmes ($\alpha \rightarrow 0$ et $\alpha \rightarrow 1$) et que l'incorporation de cumulants d'ordre de plus en plus élevé n'améliorait pas nécessairement la précision. Par ailleurs, les expressions (4.39) et (4.40) des cumulants d'un modèle linéaire avec interactions ou quadratique ne sont valables que si les variables du modèle sont normales. Concernant le coût calculatoire pour estimer les centiles d'une forme quadratique, il dépend principalement du coût pour calculer ses cumulants. Si k est le nombre de paramètres du modèle, alors le calcul des cumulants avec les expressions (4.39) et (4.40) présente une complexité en $O(k^3)$.

4.4 Application de l'analyse de variabilité aux circuits analogiques

Dans les parties précédentes, nous avons détaillé les deux points principaux de la méthode proposée pour analyser la variabilité des circuits analogiques : la modélisation des performances avec les plans d'expériences et l'approximation de Cornish-Fisher pour estimer leurs bornes de variation. Nous allons maintenant présenter les résultats obtenus sur des circuits analogiques.

4.4.1 Comparaison de méthodes

Afin de tester notre méthode, nous l'avons comparée à trois autres approches qui permettent elles aussi d'estimer les bornes de variation des performances. Les quatre approches testées sont donc les suivantes :

- l'approche traditionnelle notée **{Eldo, MC}** où une analyse de Monte Carlo est appliquée au simulateur électrique Eldo,
- l'approche notée **{modèle linéaire, $\mu \pm 3\sigma$ }** qui consiste à approximer les performances des circuits par un modèle linéaire sans interactions (plan de Plackett-Burman) et à estimer leurs bornes de variation directement à partir des expressions (4.21) et (4.22),
- l'approche notée **{modèle quadratique, MC}** où les performances des circuits sont approximées par un modèle quadratique (plan composite centré) et leurs bornes de variation estimées en ayant recours à une analyse de Monte Carlo,
- notre approche notée **{modèle quadratique, CF}** dans laquelle les performances des circuits sont également approximées par un modèle quadratique (plan composite centré), mais où leurs bornes de variation sont estimées avec les expressions (4.45) et (4.46) issues de l'approximation de Cornish-Fisher établie avec les cinq premiers cumulants.

Dans cette étude, les paramètres de conception et les paramètres environnementaux sont constants, seules les variations globales des paramètres technologiques influents ont été prises en compte. Ces variations sont modélisées par des variables aléatoires normales $N(0, \sigma)$. Le domaine d'étude choisi pour la construction des modèles est l'intervalle $[-3\sigma, +3\sigma]$ où σ est l'écart type de chaque variation. Les variations locales pourraient également être prises en compte. Cependant, étant donné que le nombre de variations locales augmente proportionnellement avec le nombre de composants dans le circuit (cf. §2.3.3), nous nous sommes dans un premier temps limités aux variations globales. Néanmoins, l'étude des variations locales serait tout à fait envisageable avec la mise en place de techniques de sélection des variations prépondérantes (plans d'expériences de criblage par exemple) afin de limiter le nombre de variations locales à prendre en compte.

Les critères utilisés pour comparer ces différentes approches sont la précision sur l'estimation des bornes et le temps de calcul. La précision est calculée à partir de l'erreur relative entre les valeurs obtenues avec les trois dernières méthodes et les valeurs de l'approche **{Eldo, MC}** qui sert de référence pour la précision. Quant au temps de calcul total, il peut se décomposer selon le temps de construction du modèle et le temps d'estimation des bornes. Les coûts de calcul des trois approches sont résumés dans le Tableau 4.12 pour des modèles comportant k paramètres. La construction d'un modèle linéaire avec un plan de Plackett-Burman requiert un nombre de simulations égal à $N_{\text{linéaire}} = (1 + \text{Ent}(k/4)) \times 4$ (où $\text{Ent}()$ est la fonction partie entière), tandis que la construction d'un modèle quadratique avec un plan composite nécessite $N_{\text{quadratique}} = 2^{k-p} + 1 + 2k$ simulations (cf. §4.2.3d). L'estimation des bornes avec l'approche **{modèle linéaire, $\mu \pm 3\sigma$ }** en utilisant les expressions (4.21) et (4.22) présente une complexité en $O(k^2)$. L'approximation de Cornish-Fisher

permet d'estimer les bornes avec une complexité en $O(k^3)$. Enfin, l'évaluation du modèle quadratique (4.24) requiert un coût de l'ordre de $O(k^2)$. Ainsi, si N_{MC} simulations Monte Carlo sont réalisées pour estimer les bornes, la complexité de l'approche **{modèle quadratique, MC}** est en $O(N_{MC} \times k^2)$. Il ressort que l'approche **{modèle linéaire, $\mu \pm 3\sigma$ }** est celle qui présente la plus faible complexité. Concernant les deux autres approches basées sur un modèle quadratique, le nombre de simulations pour construire le modèle est identique, la différence porte donc sur le coût de calcul pour estimer les bornes. Pour que l'approche **{modèle quadratique, MC}** soit précise, le nombre de simulations Monte Carlo doit être élevé, généralement très supérieur au nombre de paramètres ($N_{MC} \gg k$). Comparé à l'approche précédente, le coût calculatoire de l'approche proposée **{modèle quadratique, CF}** est donc bien plus faible, d'où son intérêt.

Tableau 4.12 : Coût calculatoire des trois approches

Approche	Nombre de simulations pour la construction du modèle	Coût du calcul d'estimation des bornes
{modèle linéaire, $\mu \pm 3\sigma$ }	$N_{linéaire} = (1 + \text{Ent}(k/4)) \times 4$	$O(k^2)$
{modèle quadratique, MC}	$N_{quadratique} = 2^{k-p} + 1 + 2k$	$O(N_{MC} \times k^2)$
{modèle quadratique, CF}	$N_{quadratique} = 2^{k-p} + 1 + 2k$	$O(k^3)$

Les circuits testés ont tous été simulés à partir du design kit d'une technologie CMOS 65nm développée par STMicroelectronics [119]. Les plans d'expériences mis en place pour construire les modèles polynomiaux des performances sont réalisés avec le simulateur électrique Eldo [38] tandis que le calcul des coefficients de chaque modèle ainsi que le calcul des cumulants et l'estimation des bornes avec le développement de Cornish-Fisher sont effectués avec Matlab [109]. Une station de travail SUN cadencée à 2.8 GHz et équipée de 2 Go de mémoire vive a été utilisée pour tester les trois approches.

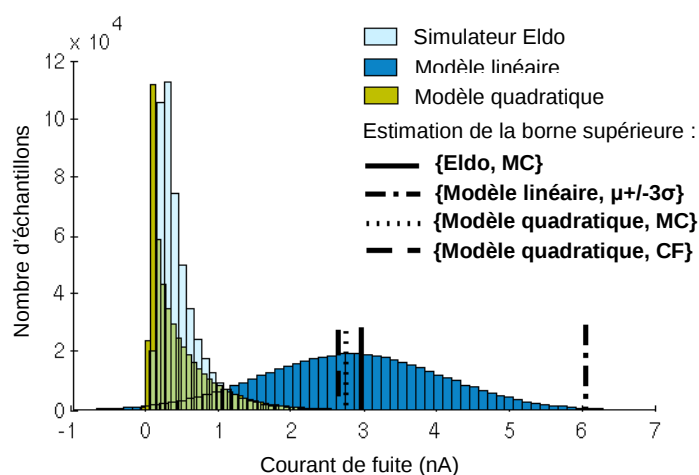
4.4.2 Porte OU exclusif

Une porte logique OU exclusif, qui présente un courant de fuite dont la distribution est typiquement non-gaussienne, constitue un circuit intéressant à tester pour montrer l'efficacité de l'approximation de Cornish-Fisher. La variabilité du courant de fuite est étudiée en fonction des variations globales de cinq paramètres technologiques. Les caractéristiques de l'étude sont résumées dans le Tableau 4.13.

Tableau 4.13 : Etude d'une porte OU exclusif

Performance	Courant de fuite
Variations globales des paramètres technologiques	Dimensions de la grille Dimensions de la zone active Epaisseur d'oxyde de grille Tension de seuil NMOS Tension de seuil PMOS
Modèle linéaire	Plan de Plackett-Burman (8 simulations)
Modèle quadratique	Plan composite centré (27 simulations)

Sur la Figure 4.11 sont illustrés les histogrammes du courant de fuite obtenus à partir des simulations Monte Carlo avec le simulateur Eldo (500 000 échantillons), le modèle linéaire et le modèle quadratique. La borne supérieure des variations (99^{ème} centile) est estimée avec les quatre approches. L'histogramme obtenu avec Eldo indique clairement que la distribution du courant de fuite est asymétrique et ne suit pas une gaussienne. Le modèle quadratique fournit ainsi une meilleure approximation de la distribution que le modèle linéaire. L'approche **{modèle linéaire, $\mu \pm 3\sigma$ }** surestime donc la borne supérieure du courant de fuite, alors que les approches **{modèle quadratique, MC}** et **{modèle quadratique, CF}** donnent une meilleure estimation.

**Figure 4.11 : Histogramme du courant de fuite**

4.4.3 Amplificateur opérationnel à transconductance

Le deuxième circuit testé est un amplificateur opérationnel à transconductance (cf. Figure 4.12). Les bornes de variation de deux performances sont estimées : le gain différentiel et la fréquence de coupure. Les caractéristiques de l'étude sont données dans le Tableau 4.14.

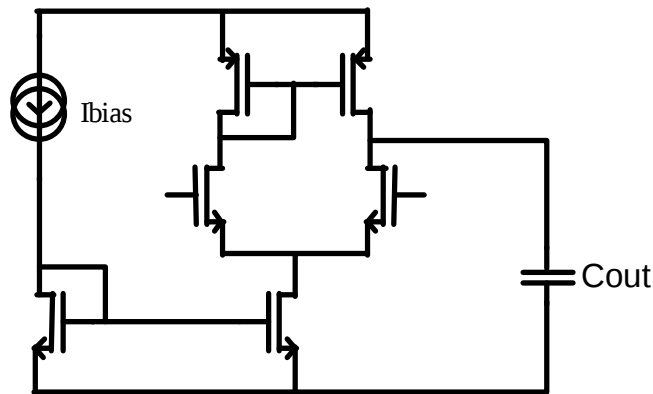


Figure 4.12 : Amplificateur opérationnel à transconductance

Tableau 4.14 : Etude d'un amplificateur opérationnel à transconductance

Performances	Fréquence de coupure Gain différentiel
Variations globales des paramètres technologiques	Dimensions de la grille Dimensions de la zone active Epaisseur d'oxyde de grille Tension de seuil NMOS Tension de seuil PMOS
Modèle linéaire	Plan de Plackett-Burman (8 simulations)
Modèle quadratique	Plan composite centré (27 simulations)

Les histogrammes de la fréquence de coupure et du gain différentiel obtenus avec l'approche **{Eldo, MC}** (10 000 échantillons) sont illustrés sur la Figure 4.13. L'erreur relative sur l'estimation des bornes de variation est donnée par la Figure 4.14. L'estimation des centiles avec l'approche **{modèle linéaire, $\mu \pm 3\sigma$ }** est la moins précise des trois approches. En revanche, l'approche **{modèle quadratique, CF}** donne des résultats aussi précis que l'approche **{modèle quadratique, MC}**.

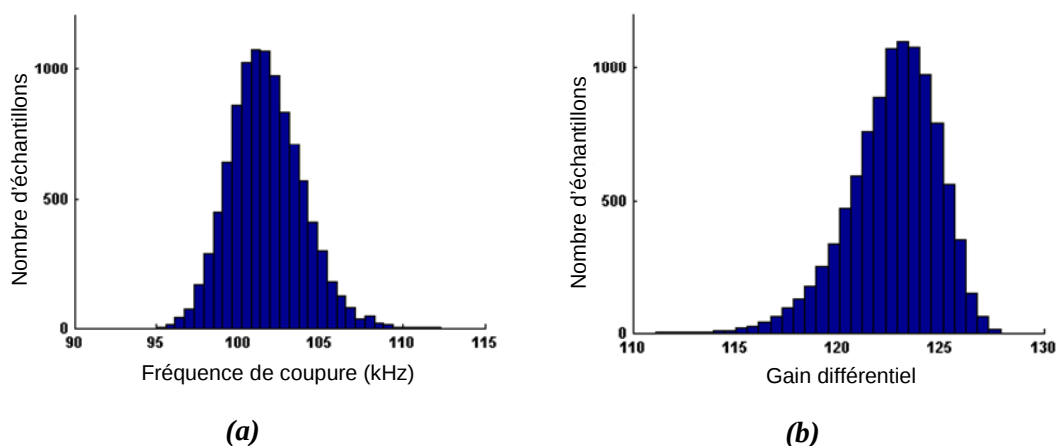


Figure 4.13 : Histogrammes de la fréquence de coupure (a) et du gain différentiel (b)

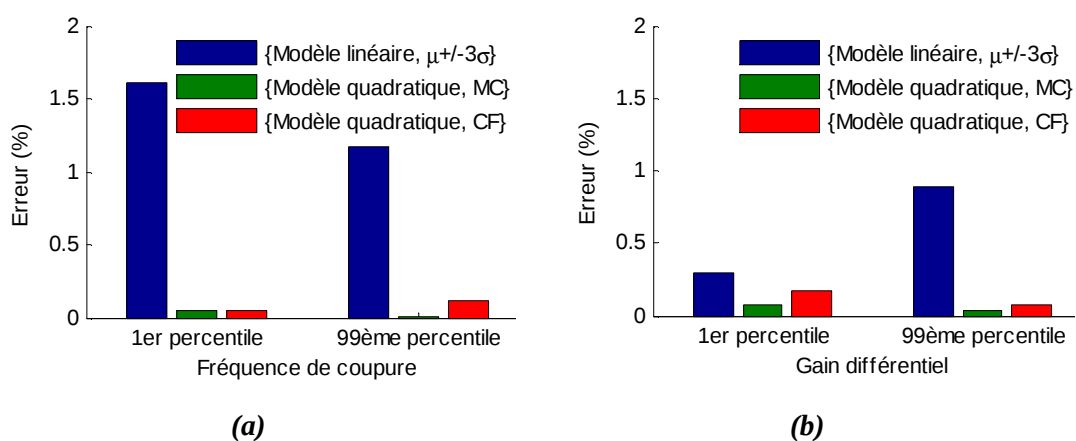


Figure 4.14 : Erreur relative sur l'estimation des bornes (1^{er} et 99^{ème} centiles) de la fréquence de coupure et du gain différentiel

Tableau 4.15 : Temps de calcul pour construire les modèles et estimer les bornes de variation de l'amplificateur opérationnel à transconductance

	{modèle linéaire, $\mu \pm 3\sigma$ }	{modèle quadratique, MC}	{modèle quadratique, CF}
Construction du modèle	1.4 s	5 s	5 s
Estimation des bornes de variation	0.25 ms	10 ms	0.7 ms

4.4.4 Amplificateur opérationnel de Miller

Le troisième circuit testé est un amplificateur opérationnel de Miller (cf. Figure 4.15). La variabilité de quatre performances est étudiée : la fréquence de coupure, le gain différentiel, la marge de phase et la consommation en courant. Le Tableau 4.16 résume les caractéristiques de l'étude.

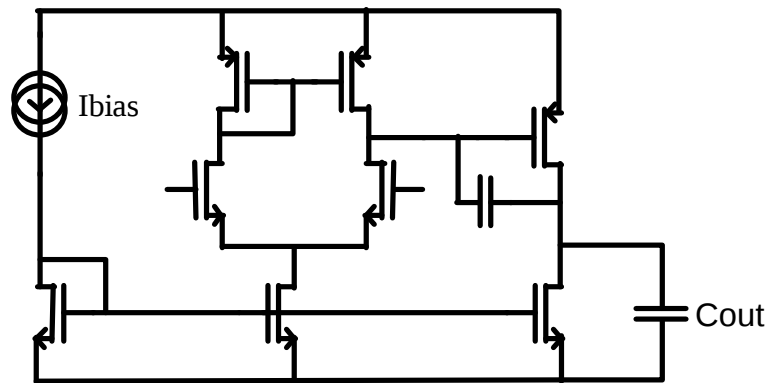


Figure 4.15 : Amplificateur opérationnel de Miller

Tableau 4.16 : Etude d'un amplificateur opérationnel de Miller

Performances	Fréquence de coupure Gain différentiel Marge de phase Consommation en courant
Variations globales des paramètres technologiques	Dimensions de la grille Dimensions de la zone active Epaisseur d'oxyde de grille Tension de seuil NMOS Tension de seuil PMOS Dimensions de capacité Poly-Poly
Modèle linéaire	Plan de Plackett-Burman (8 simulations)
Modèle quadratique	Plan composite centré (45 simulations)

Les histogrammes des quatre performances obtenus avec l'approche **{Eldo, MC}** (10 000 échantillons) sont illustrés sur la Figure 4.16 et la Figure 4.17. L'erreur relative sur l'estimation des bornes de variation est donnée par la Figure 4.18 et la Figure 4.19. La forme des distributions de la fréquence de coupure et du gain est asymétrique et ne se rapproche pas d'une gaussienne. Ces deux performances sont donc clairement non-linéaires, ce qui explique que l'estimation des bornes avec

l'approche **{modèle linéaire, $\mu \pm 3\sigma$ }** est très imprécise (20% d'erreur). En revanche, l'estimation des bornes avec le modèle quadratique permet de mieux appréhender la non-linéarité de ces deux performances et aboutit à des résultats bien plus précis comme le montrent les approches **{modèle quadratique, MC}** et **{modèle quadratique, CF}**. A l'opposé, les distributions de la marge de phase et de la consommation en courant se rapprochent plus d'une gaussienne et l'estimation des bornes avec le modèle linéaire semble suffisante.

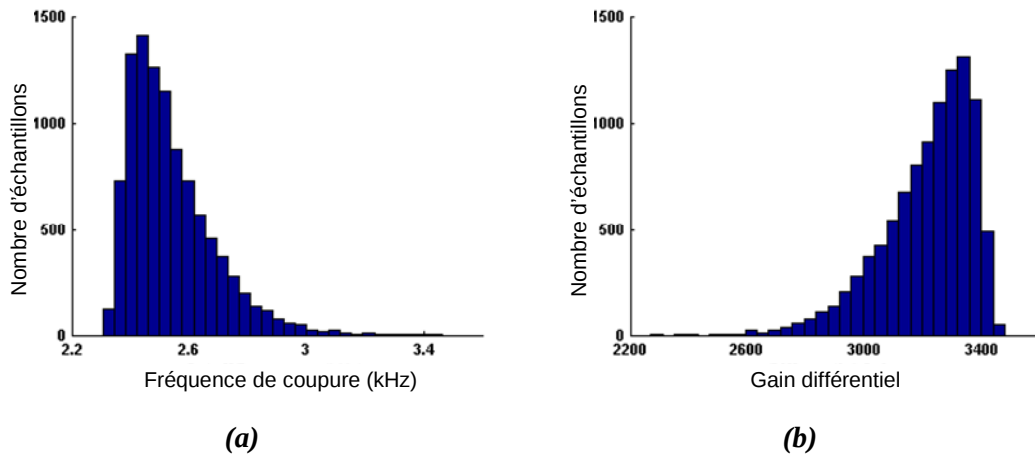


Figure 4.16 : Histogrammes de la fréquence de coupure (a) et du gain différentiel (b)

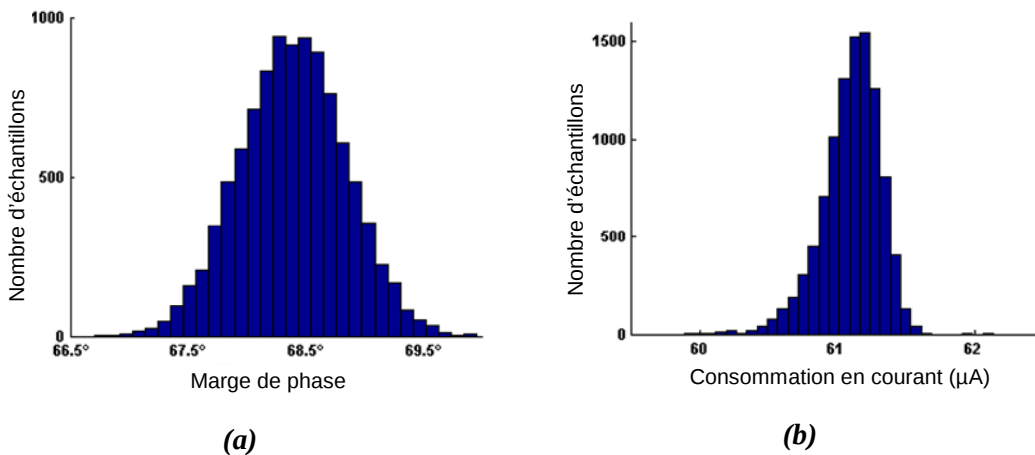


Figure 4.17 : Histogrammes de la marge de phase (a) et de la consommation de courant (b)

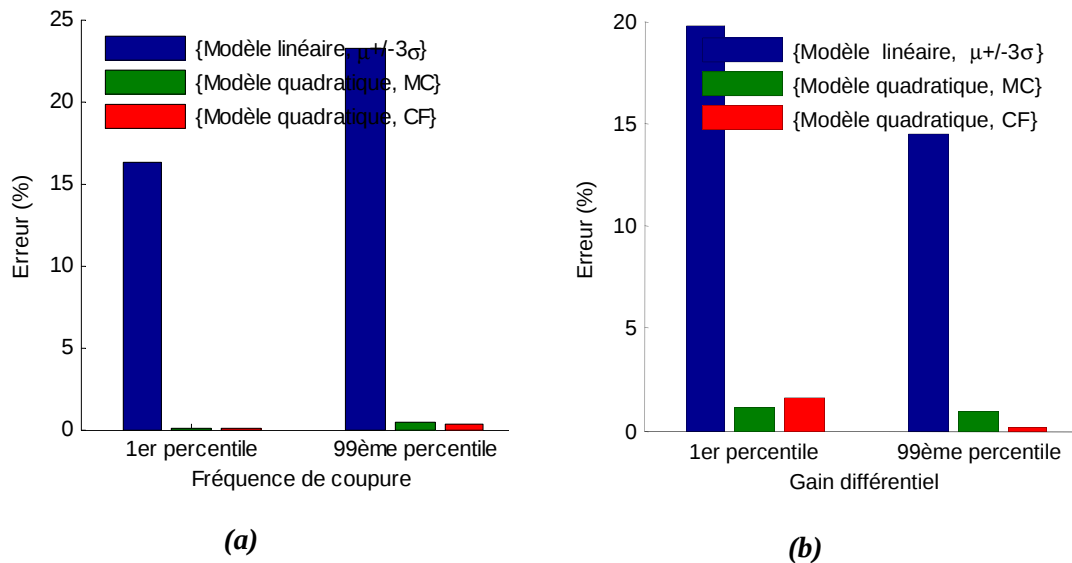


Figure 4.18 : Erreur relative sur l'estimation des bornes (1^{er} et 99^{ème} centiles) de la fréquence de coupure et du gain différentiel

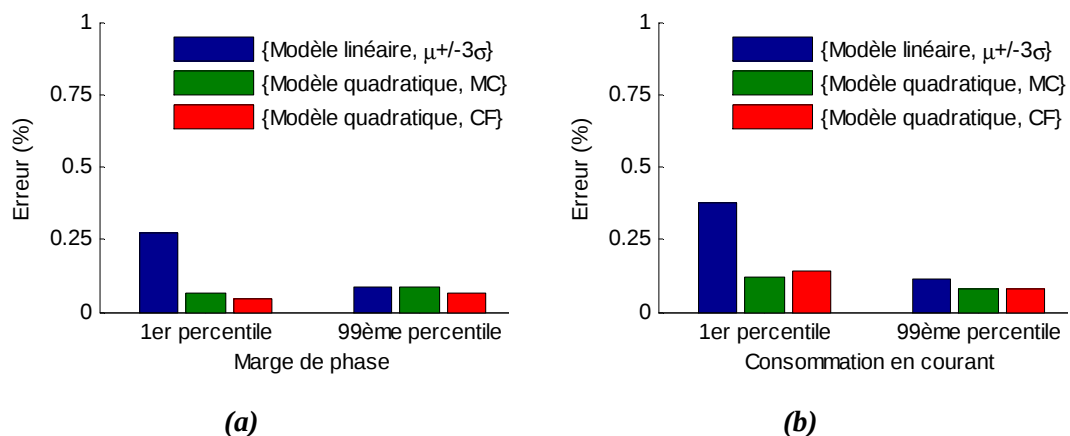


Figure 4.19 : Erreur relative sur l'estimation des bornes (1^{er} et 99^{ème} centiles) de la marge de phase et de la consommation en courant

Tableau 4.17 : Temps de calcul pour construire les modèles et estimer les bornes de variation de l'amplificateur opérationnel de Miller

	{modèle linéaire, $\mu \pm 3\sigma$ }	{modèle quadratique, MC}	{modèle quadratique, CF}
Construction du modèle	2.4 s	13.5 s	13.5 s
Estimation des bornes de variation	0.25 ms	10 ms	0.7 ms

4.4.5 Bilan

Les résultats sur les circuits testés montrent que le modèle linéaire sans interactions peut se révéler insuffisant pour estimer les bornes de variation des performances ; le modèle quadratique est alors nécessaire. En effet, les deux approches **{modèle quadratique, MC}** et **{modèle quadratique, CF}** basées sur le modèle quadratique fournissent une bien meilleure estimation des bornes. Les résultats obtenus avec ces deux approches sont quasiment identiques en matière de précision, cependant elles se différencient par leur temps de calcul. Les temps de construction des modèles polynomiaux et d'estimation des bornes de variation avec chaque approche sont donnés dans le Tableau 4.15 pour l'amplificateur opérationnel à transconductance et dans le Tableau 4.17 pour l'amplificateur opérationnel de Miller. Deux points importants sont à noter. D'une part, le temps d'estimation des bornes est très faible devant le temps de construction des modèles. Ceci s'explique par le temps de calcul des simulations Eldo lors de l'exécution du plan d'expériences. D'autre part, l'approche **{modèle quadratique, CF}** est dix fois plus rapide pour estimer les bornes que l'approche **{modèle quadratique, MC}** pour un niveau de précision équivalent. Même si ces temps sont très faibles devant les temps de construction du modèle, le facteur 10 entre les deux méthodes peut devenir significatif si l'estimation des bornes est réitérée plusieurs fois dans un algorithme d'optimisation par exemple.

4.5 Conclusion

Une nouvelle méthode pour analyser la variabilité des circuits analogiques a été présentée. Son but est d'estimer les valeurs extrêmes des performances dues aux variations des paramètres technologiques. Elle se déroule en deux étapes : les performances sont d'abord approximées par des modèles linéaires avec interactions ou des modèles quadratiques grâce aux plans d'expériences, puis les valeurs pire-cas sont estimées avec le développement limité de Cornish-Fisher. Le choix des modèles polynomiaux s'explique pour plusieurs raisons : ils sont faciles à construire et aisément interprétables, de plus ils se révèlent suffisants pour des approximations locales. Pour construire ces méta-modèles, les plans d'expériences ont été utilisés afin d'obtenir la meilleure précision possible avec un nombre de simulations minimum. Cependant, leur coût de construction devient prohibitif au-delà de vingt facteurs en jeu, ce qui limite leur application à des circuits analogiques faiblement complexes. Enfin l'approximation de Cornish-Fisher fournit une expression analytique précise et efficace pour estimer les bornes de variation. La méthode a été testée sur des circuits analogiques basiques. Les résultats montrent, premièrement, que les modèles quadratiques permettent de bien approximer les non-linéarités des performances, et deuxièmement, que le développement de Cornish-Fisher permet de réduire le temps d'estimation des bornes d'un facteur dix par rapport à une

analyse de Monte Carlo pour une précision aussi bonne. Plus généralement, comparée à l'analyse pire-cas et l'analyse analytique présentées dans le Chapitre 3, la méthode basée sur l'approximation de Cornish-Fisher offre une bien meilleure précision, tandis qu'elle requiert un coût de calcul très inférieur à celui de l'approche Monte Carlo. Cependant, comme pour la méthode APEX, la méthode proposée nécessite que les variations paramétriques soient modélisées par des variables aléatoires normales. Grâce à ses très bonnes caractéristiques aussi bien en termes de précision que de coût de calcul, cette approche convient parfaitement à l'estimation des performances pire-cas au sein d'un algorithme d'optimisation robuste.

Chapitre 5 : Modélisation par morceaux des performances des circuits analogiques

Dans le Chapitre 4, nous avons présenté une méthode d'approximation des performances basée sur les plans d'expériences. Grâce à cette méthode, nous avons pu construire un modèle des performances en fonction des variations des paramètres technologiques et estimer les valeurs pire-cas des performances. Dans le présent chapitre, nous allons étendre cette méthode d'approximation des performances aux paramètres de conception et aux paramètres environnementaux en mettant en place une modélisation par morceaux. Cette méthode va nous permettre d'approximer les performances des circuits en fonction des différents types de paramètres, sur des domaines relativement larges pour lesquels les modèles quadratiques se révèlent insuffisants. Les aspects importants de la méthode seront tout d'abord passés en revue, puis les résultats obtenus sur des circuits analogiques seront présentés. La méthode de modélisation par morceaux présentée s'avère particulièrement efficace pour approcher la forme générale des performances pour un coût de calcul réduit.

5.1 Introduction

La réduction des dimensions dans les technologies CMOS a un double impact sur la variabilité des circuits. D'une part, le nombre de sources de variation augmente, avec notamment l'émergence des fluctuations aléatoires liées au dopage (cf. Chapitre 2). D'autre part, l'amplitude des variations tend à s'accroître, ce qui accentue leur influence sur les performances. De plus, le comportement des circuits dépend considérablement des zones de fonctionnement des transistors et est extrêmement non-linéaire entre ces zones. L'approximation des performances avec un modèle quadratique peut alors se révéler insuffisante pour modéliser toutes ces non-linéarités. Le recours à des polynômes de degré plus élevé peut sembler être la solution. Cependant, augmenter le degré des polynômes conduit à augmenter le nombre de coefficients et donc le nombre de simulations dans le plan d'expériences : un modèle linéaire avec k paramètres comporte $1+k$ coefficients, un modèle quadratique $1+2k+k(k-1)/2$ et un modèle cubique complet $1+3k+3k(k-1)/2+k(k-1)(k-2)/6$. La méthode de modélisation la plus appropriée doit donc à la fois fournir une bonne approximation de la non-linéarité des performances et avoir un faible coût de construction. Les différents modèles de polynômes ainsi que cette modélisation « idéale » sont représentés schématiquement sur la Figure 5.1.

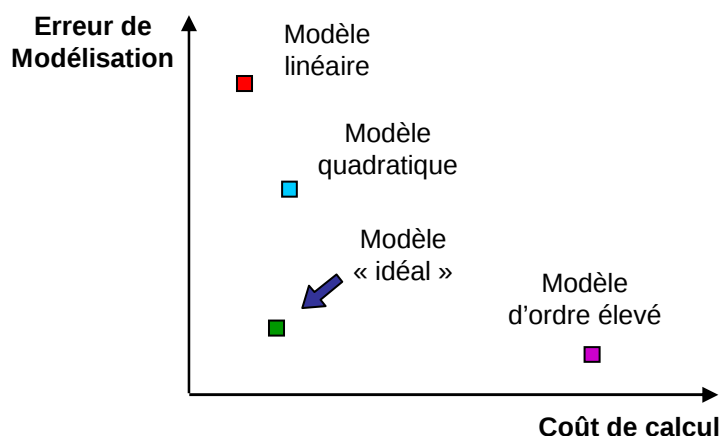


Figure 5.1 : Comparaison des modèles polynomiaux en fonction de la précision et du coût de calcul

Pour atteindre ce compromis entre précision et coût de calcul, nous avons donc développé une méthode d'approximation des performances par des modèles linéaires ou quadratiques par morceaux : le domaine d'étude est divisé en plusieurs sous-domaines sur lesquels un polynôme d'ordre un ou deux approxime localement la performance étudiée. La Figure 5.2 illustre le principe de la modélisation par morceaux. Cette approche présente un double intérêt en termes de modélisation globale et locale : le découpage en sous-domaines permet de couvrir l'ensemble du domaine d'étude, tandis que les modèles linéaires ou quadratiques approximent localement les non-linéarités avec précision. La modélisation par morceaux va nous permettre de construire des modèles de performances en fonction des paramètres de conception, des paramètres technologiques et des paramètres environnementaux.

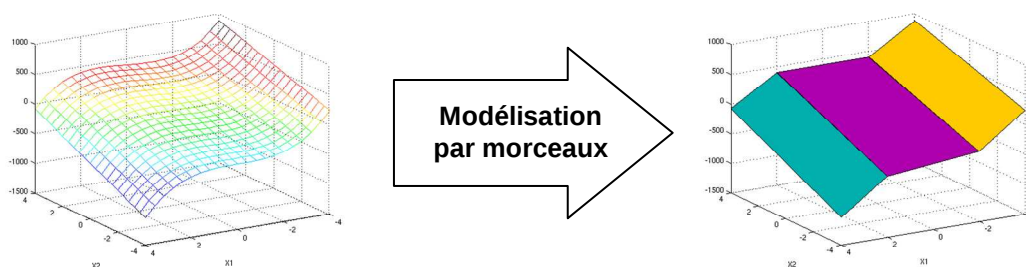


Figure 5.2 : Principe de la modélisation par morceaux

5.2 Méthode de modélisation par morceaux des performances

Dans cette partie, nous allons tout d'abord présenter l'algorithme de la méthode, puis ses points importants, à savoir : la construction séquentielle des modèles, leur validation, la stratégie de bisection et la sauvegarde des simulations.

5.2.1 Algorithme de modélisation par morceaux

L'algorithme de la méthode est représenté sur la Figure 5.3. L'objectif est de diviser le domaine d'étude en sous-domaines (aussi appelés sous-pavés, cf. Annexe C) de taille variable sur lesquels la performance à modéliser va être localement approximée par un modèle linéaire avec interactions ou un modèle quadratique. Une liste L est utilisée pour sauvegarder les différents sous-domaines générés ; au départ, cette liste contient uniquement le domaine d'étude initial. A chaque itération de l'algorithme, un sous-domaine est retiré de la liste et un modèle linéaire ou quadratique est construit sur ce sous-domaine. La précision du modèle construit est testée ; si elle est suffisante, le couple (sous-domaine, modèle) est déplacé dans une liste résultat R . Dans le cas contraire, le sous-domaine est divisé en deux nouveaux sous-domaines qui sont placés dans la liste L pour être traités ultérieurement. L'algorithme se termine lorsque la liste L est vide ; le domaine d'étude a alors été divisé en plusieurs sous-domaines sauvegardés dans la liste R . L'ensemble des modèles générés sur chaque sous-domaine de la liste R forme un modèle par morceaux des performances. Les points clés de la méthode sont détaillés dans les parties qui suivent.

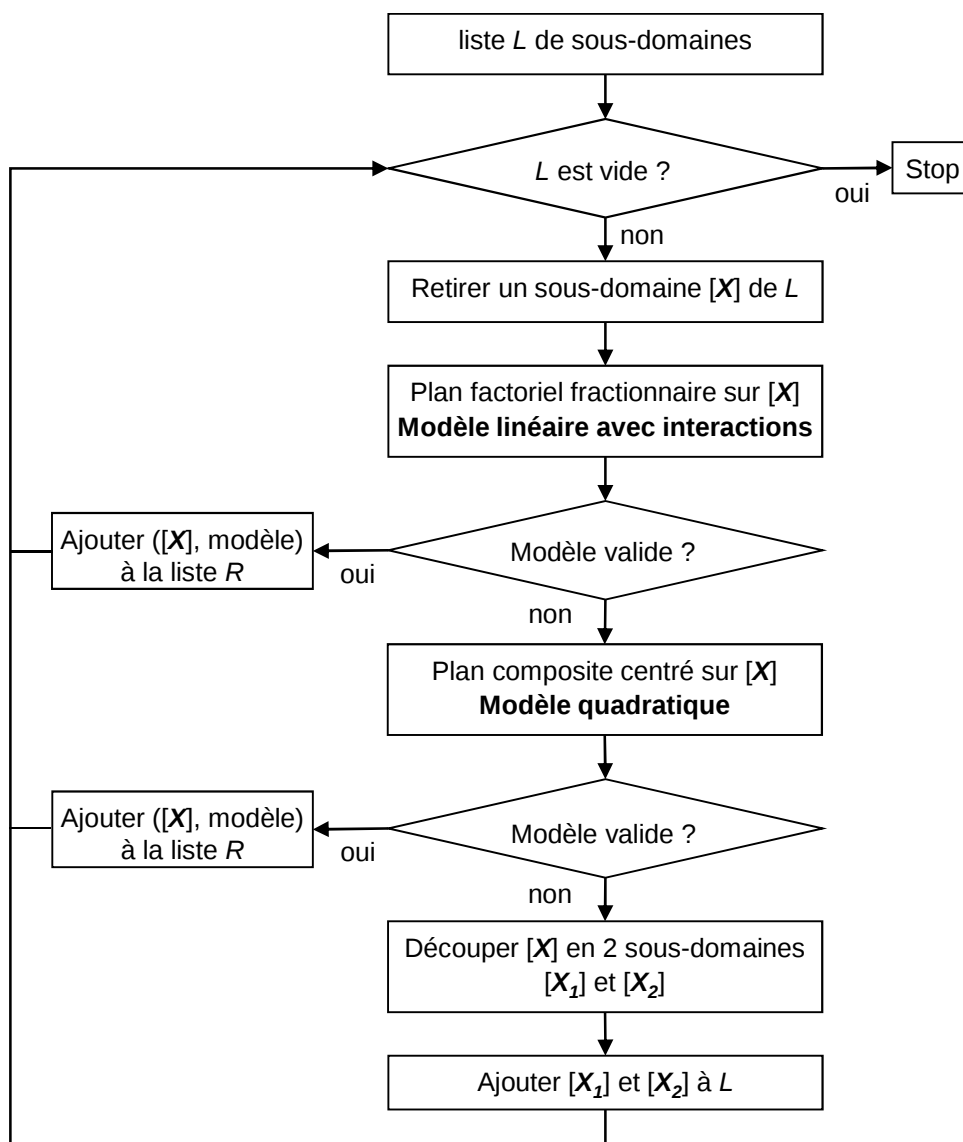


Figure 5.3 : Algorithme de modélisation par morceaux des performances

5.2.2 Modélisation séquentielle

Sur chaque sous-domaine, les modèles sont construits de façon séquentielle. Un plan factoriel fractionnaire de résolution V est, dans un premier temps, mis en place pour approximer la performance par un modèle linéaire avec les interactions d'ordre 1. Si la précision de ce modèle est suffisante pour prédire les valeurs de la performance, le modèle est sauvegardé et un nouveau sous-domaine est retiré de la liste L . Sinon, dans le cas où le modèle linéaire avec interactions n'est pas valide, le plan factoriel est alors complété par des points en étoile augmentés d'un point au centre pour obtenir un plan composite centré avec lequel un modèle quadratique va pouvoir être construit. Ainsi, avec cette approche séquentielle, le nombre de simulations nécessaires à chaque étape est minimum et fonction du type de méta-modèle. Le coût de calcul est donc adapté au plus juste au degré de non-linéarité de la performance : si cette dernière est faiblement non-linéaire sur le sous-

domaine, un modèle linéaire avec interactions sera suffisant, sinon un modèle quadratique sera bâti à moindre coût en réutilisant les points simulés pour le modèle linéaire et en simulant uniquement les points en étoile plus un point au centre.

5.2.3 Validation du modèle

Comme nous l'avons vu au Chapitre 4, la précision des modèles est testée en calculant l'erreur quadratique moyenne $RMSE$ à partir d'essais supplémentaires qui sont différents des essais du plan d'expériences. L'emplacement de ces points dans le cas d'un modèle linéaire avec interactions ou quadratique a été vu au §4.2.3c) du Chapitre 4. Une fois l'erreur quadratique moyenne calculée, sa valeur est comparée avec une valeur seuil $RMSE_{max}$ fixée par l'utilisateur. Si la valeur de l'erreur quadratique moyenne est inférieure à $RMSE_{max}$, le modèle est considéré suffisamment précis et est validé, sinon il est rejeté. Ainsi, la précision de la modélisation par morceaux et le coût de calcul peuvent être ajustés en modifiant la valeur de $RMSE_{max}$: la précision du modèle sera d'autant meilleure que la valeur de $RMSE_{max}$ sera faible, en contrepartie le nombre de bisections à effectuer sera plus grand et le coût calculatoire plus élevé.

5.2.4 Stratégie de bisection

Dans l'algorithme de modélisation proposé, si le modèle quadratique n'est pas valide, le sous-domaine est alors divisé en deux. Plusieurs possibilités sont envisageables pour déterminer la direction de bisection : au hasard ou encore en sélectionnant la direction pour laquelle l'intervalle de valeurs est le plus grand. Cependant, il est judicieux de choisir la direction qui va permettre de limiter le nombre de bisections et ainsi réduire le coût de calcul. Le critère de bisection qui a été retenu est donc basé sur le gradient du modèle quadratique ; il a été emprunté aux algorithmes d'optimisation par intervalles [120].

Soit f une fonction quadratique définie sur un domaine $D = [\underline{X}_1, \overline{X}_1] \times [\underline{X}_2, \overline{X}_2] \times \dots \times [\underline{X}_k, \overline{X}_k]$. La bisection est alors réalisée perpendiculairement à la direction i qui maximise :

$$\left[\overline{X}_i - \underline{X}_i \right] \left[\max_{\mathbf{X} \in D} (g_i(\mathbf{X})) - \min_{\mathbf{X} \in D} (g_i(\mathbf{X})) \right], \quad i = 1 \dots k \quad (5.1)$$

où g_i est la i -ème composante du gradient de f :

$$\nabla f = \begin{pmatrix} g_1 \\ g_2 \\ \dots \\ g_k \end{pmatrix} \quad (5.2)$$

Etant donné que f est une fonction quadratique, alors les g_i sont des fonctions linéaires facilement calculables. La direction qui est choisie est donc celle pour laquelle la variation du gradient est maximale. En effet plus la variation du gradient est grande, plus la fonction tend à être non-linéaire. Ainsi, découper le domaine perpendiculairement à cette direction permet d'obtenir deux sous-domaines sur lesquels la fonction est moins non-linéaire et plus facilement approximable par un modèle linéaire ou quadratique.

Nous allons considérer l'exemple suivant afin d'illustrer cette stratégie de bisection. Soit la fonction h représentée sur la Figure 5.4(a). Cette fonction est approximée par un modèle quadratique f obtenu avec un plan composite centré et représenté sur la Figure 5.4(b). Manifestement, le modèle quadratique ne fournit pas une bonne approximation de la fonction h ; le domaine doit alors être divisé en deux sous-domaines. La direction de bisection qui est retenue dépend de la variation des composantes g_1 et g_2 du gradient de f qui sont tracées sur la Figure 5.5. Il apparaît clairement que g_1 varie beaucoup plus que g_2 . D'après le critère de bisection (5.1), le domaine doit donc être découpé perpendiculairement à la direction de X_1 . Le modèle par morceaux de h construit en découpant le domaine perpendiculairement à la direction de X_1 est illustré sur la Figure 5.6(a) : il donne une bien meilleure approximation de la fonction h que le modèle quadratique initial f . La Figure 5.6(b) représente le modèle par morceaux de h si la direction perpendiculaire à X_2 avait été choisie pour diviser le domaine : il est identique au modèle quadratique f . Ainsi, même en découpant indéfiniment le domaine dans cette direction, la précision du modèle par morceaux ne s'améliorerait pas.

Le critère de bisection (5.1) basé sur les variations du gradient permet donc à la fois :

- d'améliorer la précision en déterminant efficacement selon quelle direction découper les sous-domaines,
- de réduire le coût de calcul en limitant le nombre de bisections.

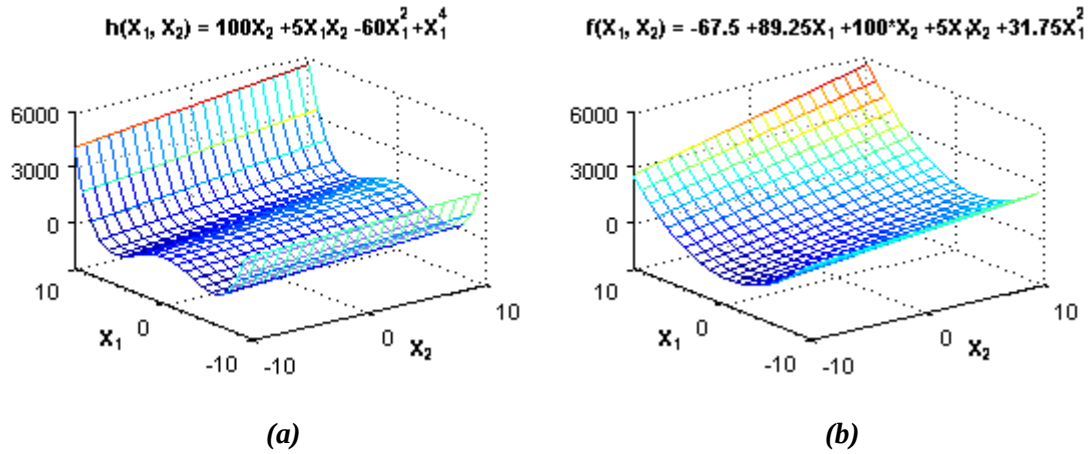


Figure 5.4 : Représentation graphique de la fonction h (a) et de son approximation quadratique f (b)

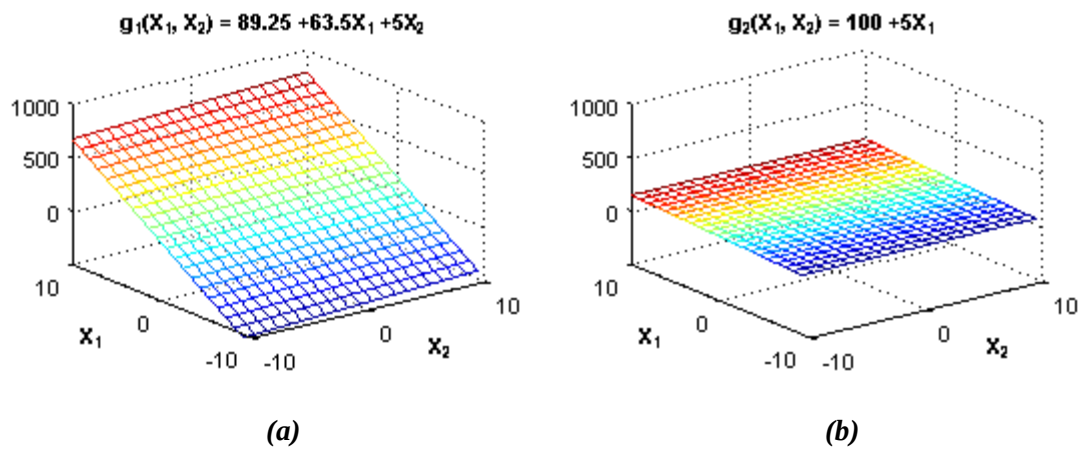


Figure 5.5 : Représentation graphique des composantes g_1 (a) et g_2 (b) du gradient de l'approximation quadratique f

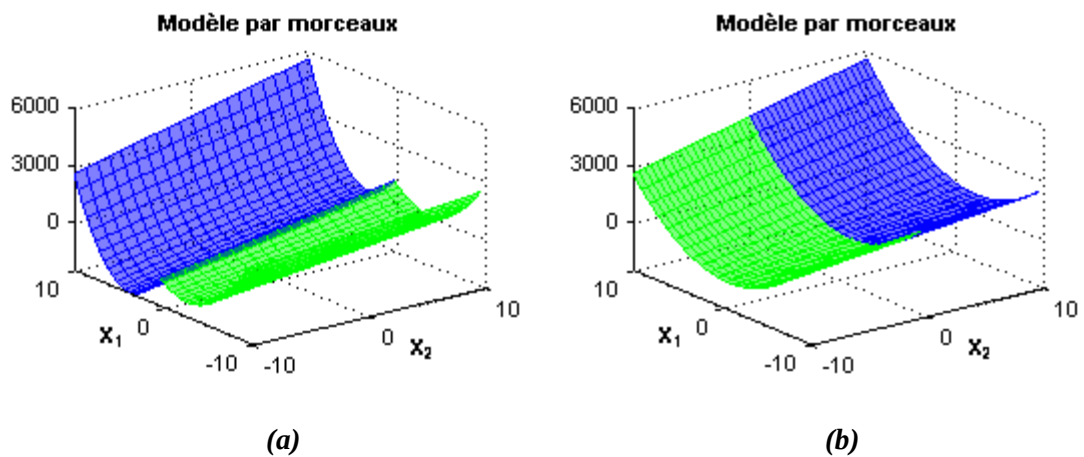


Figure 5.6 : Modèles par morceaux de h en découpant le domaine initial perpendiculairement à la direction de X_1 (a) et perpendiculairement à la direction de X_2 (b)

5.2.5 Sauvegarde des points

Tous les points de simulation et les valeurs associées de la performance sont sauvegardés de façon à pouvoir être réutilisés par la suite. En effet, si le modèle quadratique n'est pas valide, le domaine est divisé en deux, et sur chaque sous-domaine, un certain nombre de points appartenant au domaine initial peuvent être réutilisés : les sauvegarder permet d'économiser autant de simulations et donc de réduire le coût de calcul. Les points simulés pour construire un modèle quadratique qui dépend de trois facteurs sont illustrés sur la Figure 5.7(a) : les points du plan composite centré sont en rouge, tandis que les points pour tester le modèle sont en bleu. Sur la Figure 5.7(b) sont représentés les points qui appartiennent au sous-domaine de « gauche » et qui peuvent être réutilisés après avoir découpé le domaine initial perpendiculairement à la direction de X_1 . Les points en vert correspondent aux points qui vont pouvoir être réutilisés dans un plan factoriel complet pour construire un modèle linéaire avec interactions sur le sous-domaine : ils représentent la moitié des points du plan factoriel. L'autre moitié du plan est représentée par les points en jaune sur la Figure 5.7(b). Ainsi, en sauvegardant le résultat des points de simulation, seule la moitié des points du plan factoriel devra être simulé ; le temps de simulation est donc divisé par deux par rapport à une modélisation sans sauvegarde. Enfin, les points en violet clair et en violet foncé sont respectivement les points en étoile et les points de test du modèle quadratique sur le domaine initial et qui appartiennent au sous-domaine de « gauche ». Ces points vont également être réutilisés afin d'enrichir le plan factoriel. En effet, avec plus de points dans le plan d'expériences, l'estimation des coefficients du modèle sera meilleure. L'exemple présenté est basé sur un plan factoriel complet, cependant les avantages obtenus avec la sauvegarde des points restent valables pour un plan factoriel fractionnaire. Ainsi, la sauvegarde des points dans l'approche de modélisation par morceaux permet à la fois de réduire le coût calculatoire et d'améliorer la précision du modèle.

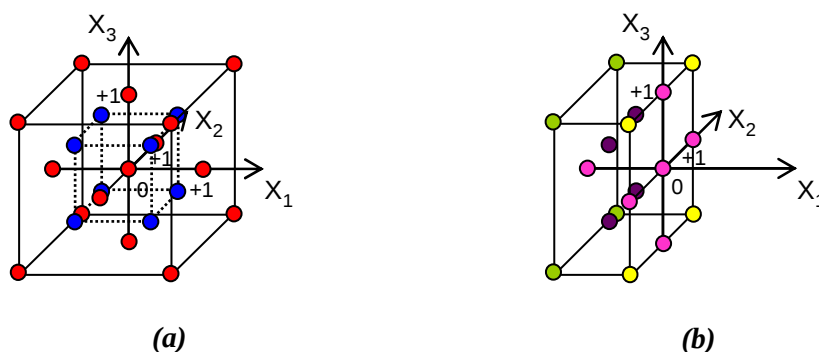


Figure 5.7 : Points du plan composite centré simulés sur le domaine initial (a) et réutilisés sur le sous-domaine de « gauche » (b).

5.2.6 Exemple numérique

Pour illustrer l'algorithme de modélisation par morceaux, nous avons repris l'exemple traité au §5.2.4. La fonction h à approximer est représentée sur la Figure 5.8(a). Les modèles construits à chaque itération de l'algorithme sont résumés dans le Tableau 5.1 et illustrés sur les Figure 5.8(b) à 5.8(h). Le domaine initial a été divisé en quatre sous-domaines : deux modèles quadratiques et deux modèles linéaires ont été construits pour approximer par morceaux la fonction h .

Tableau 5.1 : Modèle construit à chaque itération de l'algorithme de modélisation par morceaux

Itération	Modèle	Sous-domaine		Figure
		X_1	X_2	
1	Modèle quadratique non-valide	[-9 10]	[-10 10]	Figure 5.8(b)
2	Modèle quadratique non-valide	[-9 0.5]	[-10 10]	Figure 5.8(c)
3	Modèle quadratique valide	[-9 -4.25]	[-10 10]	Figure 5.8(d)
4	Modèle linéaire avec interactions valide	[-4.25 0.5]	[-10 10]	Figure 5.8(e)
5	Modèle quadratique non-valide	[0.5 10]	[-10 10]	Figure 5.8(f)
6	Modèle linéaire avec interactions valide	[0.5 5.25]	[-10 10]	Figure 5.8(g)
7	Modèle quadratique valide	[5.25 10]	[-10 10]	Figure 5.8(h)

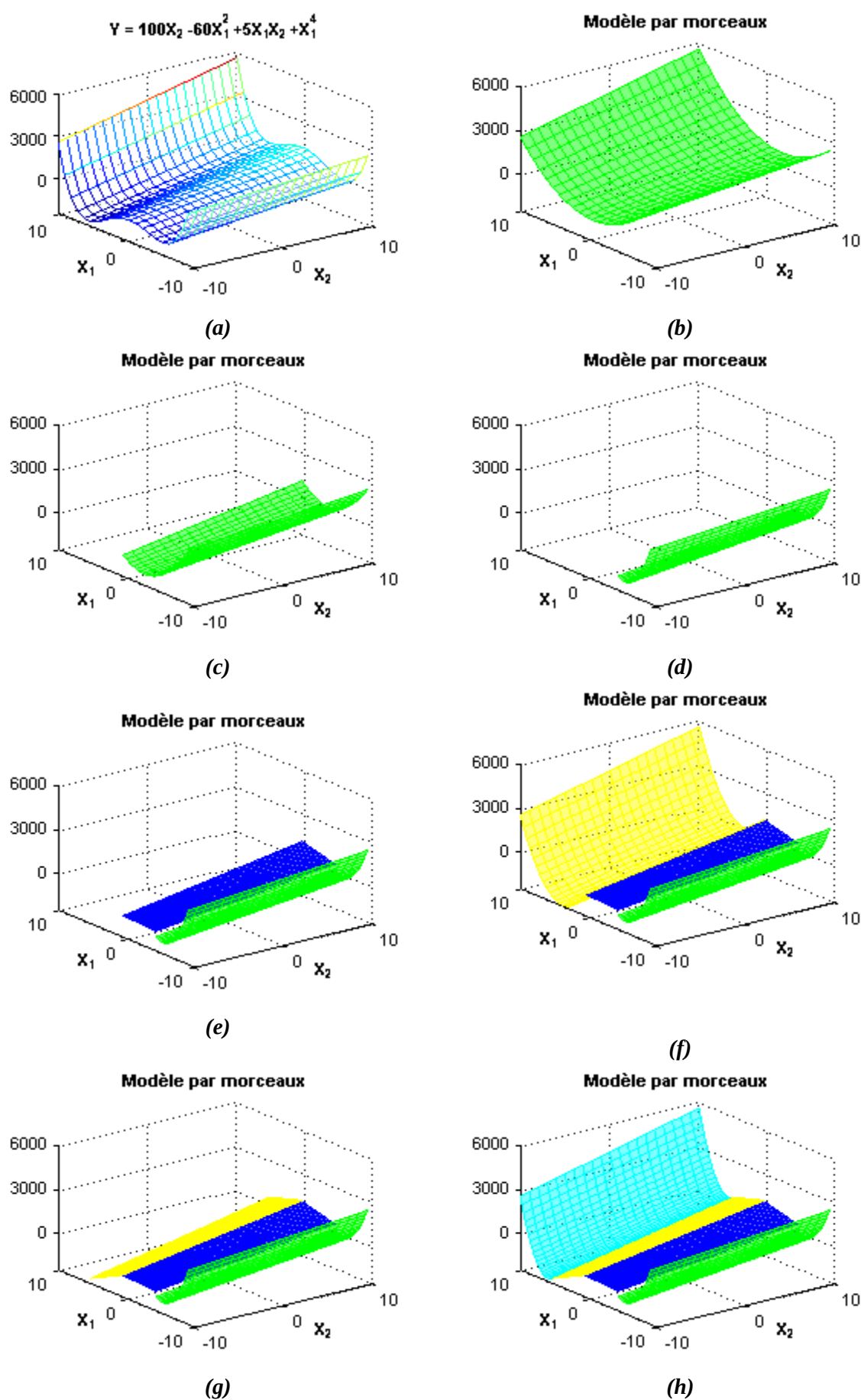


Figure 5.8 : Modèles construits à chaque itération de l'algorithme de modélisation par morceaux

5.3 Application à la modélisation des performances d'un amplificateur opérationnel à transconductance

La modélisation par morceaux a été mise en œuvre pour approximer les performances d'un amplificateur opérationnel à transconductance illustré sur la Figure 5.9 où $(W/L)_i$ représente le ratio entre la largeur W_i et la longueur L_i du canal pour chaque transistor. Les deux performances approximées sont le gain différentiel et la fréquence de coupure.

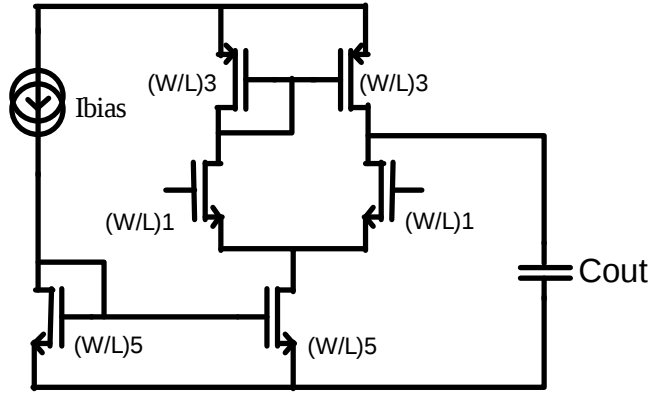


Figure 5.9 : Amplificateur opérationnel à transconductance

5.3.1 Comparaison de méthodes

Pour chaque performance, nous avons comparé la modélisation par morceaux avec une modélisation quadratique et une modélisation cubique obtenues grâce à des plans d'expériences D-Optimaux [103]. L'intérêt de ces plans est d'être les plus économes en termes d'essais pour un modèle donné puisqu'il y a autant d'essais que de coefficients dans le modèle. Il faut cependant souligner que la construction d'un plan D-Optimal nécessite la résolution d'un problème d'optimisation, ce qui peut prendre un certain temps, notamment dans le cas du modèle cubique où le nombre de coefficients est élevé. Les plans D-Optimaux pour le modèle quadratique et le modèle cubique sont générés grâce au logiciel Design-Expert [110]. Les expressions du modèle quadratique et du modèle cubique sont respectivement données par les formules (5.3) et (5.4) où k est le nombre de paramètres.

$$\hat{Y}_{\text{quadratique}} = a_0 + \sum_{i=1}^k b_i X_i + \sum_{i=1}^{k-1} \sum_{q=i+1}^k c_{iq} X_i X_q + \sum_{i=1}^k c_i X_i^2 \quad (5.3)$$

$$\hat{Y}_{\text{cubique}} = \hat{Y}_{\text{quadratique}} + \sum_{i=1}^{k-2} \sum_{q=i+1}^{k-1} \sum_{r=q+1}^k d_{iqr} X_i X_q X_r + \sum_{i=1}^{k-1} \sum_{q=i+1}^k e_{iq} X_i^2 X_q + \sum_{i=1}^{k-1} \sum_{q=i+1}^k f_{iq} X_i X_q^2 + \sum_{i=1}^k g_i X_i^3 \quad (5.4)$$

Les critères de comparaison sont la précision de chaque modèle et le coût calculatoire (nombre de simulations) pour le construire. Le nombre de simulations tient compte à la fois des simulations pour construire le modèle, mais également des simulations supplémentaires pour tester sa validité. La précision de chaque modèle, quant à elle, est estimée à partir du coefficient R^2 calculé sur un échantillon de 1000 points répartis aléatoirement sur le domaine initial et différents des points ayant servi à la construction du modèle :

$$R^2 = 1 - \frac{\sum_{i=1}^N (y_i - \hat{y}_i)^2}{\sum_{i=1}^N (y_i - \bar{y})^2} \quad (5.5)$$

où $\hat{y}_i$ est la valeur prédite par le modèle, y_i la valeur simulée (réelle), $\bar{y}$ la moyenne des valeurs réelles et N le nombre d'essais de l'échantillon, c'est-à-dire 1000. Plus le coefficient R^2 tend vers 1, meilleure est la précision du modèle.

5.3.2 Variables abstraites

Les paramètres du circuit pris en compte dans la modélisation ainsi que leur domaine de variation sont résumés dans le Tableau 5.2.

Tableau 5.2 : Paramètres de l'amplificateur opérationnel à transconductance

Type de paramètres	Nom	Domaine de variation
Paramètres de conception	W_1	[1μm, 50μm]
	L_1	[1μm, 5μm]
	W_3	[1μm, 50μm]
	L_3	[1μm, 5μm]
	W_5	[1μm, 50μm]
	L_5	[1μm, 5μm]
	I_{bias}	[10μA, 30μA]
Variations des paramètres technologiques (variations globales)	Dimensions de la grille	$[-3\sigma, 3\sigma]$
	Dimensions de la zone active	$[-3\sigma, 3\sigma]$
	Epaisseur d'oxyde de grille	$[-3\sigma, 3\sigma]$
	Tension de seuil NMOS	$[-3\sigma, 3\sigma]$
	Tension de seuil PMOS	$[-3\sigma, 3\sigma]$
Paramètres environnementaux	Tension d'alimentation	[2.25V, 2.75V]
	Température	[0°C, 40°C]

Cependant les performances ne sont pas exprimées directement en fonction des largeurs W_i et des longueurs L_i des transistors, mais en fonction des variables abstraites définies dans le Tableau 5.3. En effet, les performances des circuits dépendent généralement de l'intensité du courant qui traverse les transistors qui lui-même dépend du ratio entre leur largeur et leur longueur. L'expression du courant de saturation dans les transistors NMOS à canal long illustre cette dépendance :

$$I_{ds} = \frac{1}{2} K_n \frac{W}{L} (V_{GS} - V_T)^2 (1 + \lambda V_{DS}) \quad (5.6)$$

où W et L sont respectivement la largeur et la longueur du canal pour un transistor NMOS, V_{GS} la tension entre la grille et la source, V_{DS} la tension entre le drain et la source et V_T la tension de seuil, tandis que K_n et λ sont des paramètres technologiques. Ainsi, approximer les performances des circuits en fonction des variables abstraites A_i et B_i permet d'améliorer la qualité de la modélisation en construisant un modèle qui ajustera mieux la forme des performances.

Tableau 5.3 : Variables abstraites

Variable abstraite	Domaine de variation
$A_1 = \frac{W_1}{L_1}$	[1, 10]
$B_1 = L_1$	[1 μ m, 5 μ m]
$A_3 = \frac{W_3}{L_3}$	[1, 10]
$B_3 = L_3$	[1 μ m, 5 μ m]
$A_5 = \frac{W_5}{L_5}$	[1, 10]
$B_5 = L_5$	[1 μ m, 5 μ m]

5.3.3 Résultats

La méthode de modélisation par morceaux a été implémentée en Java. Une station de travail SUN équipée d'un processeur à 2.8 GHz et de 2 Go de mémoire vive a été utilisée pour tester les trois approches. Les temps de simulation pour construire les modèles sont donnés dans le Tableau 5.4. La Figure 5.10 et la Figure 5.11 comparent les trois modélisations en termes de précision et de coût calculatoire.

Tableau 5.4 : Temps de simulation pour construire les modèles

	Gain différentiel	Fréquence de coupure	
		$RMSE_{max} = 7.5 \text{ kHz}$	$RMSE_{max} = 5 \text{ kHz}$
Modèle quadratique	25 s	25 s	25 s
Modèle cubique	2 min 12 s	2 min 12 s	2 min 12 s
Modèle par morceaux	1 min 57 s	1 min 57 s	7 min 43 s

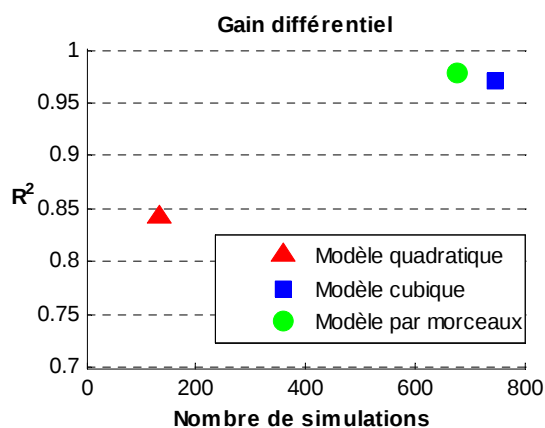


Figure 5.10 : Comparaison des trois modélisations appliquées au gain différentiel

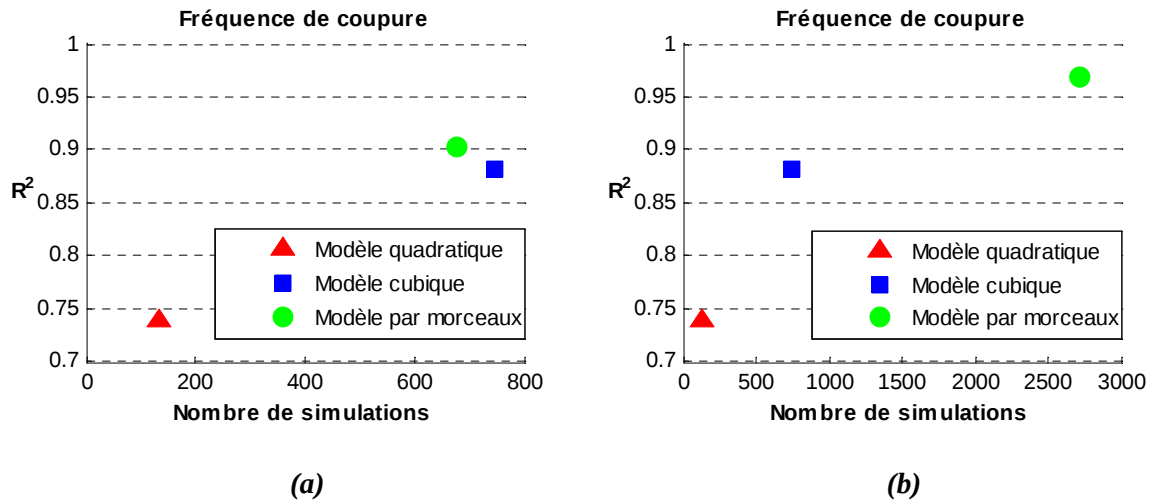


Figure 5.11 : Comparaison des trois modélisations appliquées à la fréquence de coupure pour deux valeurs différentes de $RMSE_{max}$: 7.5 kHz (a) et 5 kHz (b)

Comme prévu, le modèle quadratique est le modèle qui requiert le moins de simulations, mais qui est également le moins précis. La modélisation cubique permet, quant à elle, d'améliorer la précision par rapport au modèle quadratique au prix d'un nombre plus élevé de simulations. Par ailleurs, la création d'un plan D-Optimal pour le modèle cubique avec le logiciel Design-Expert nécessite presque une 1h de calcul en plus des simulations électriques pour construire le modèle ; le coût calculatoire de la modélisation cubique dans son ensemble est donc relativement élevé.

Contrairement aux deux approches précédentes, l'approximation par morceaux offre la possibilité de pouvoir ajuster le couple précision/coût de calcul grâce à la valeur seuil $RMSE_{max}$, comme le montrent les résultats obtenus sur la fréquence de coupure. Pour une valeur de $RMSE_{max}$ égale à 7.5 kHz (cf. Figure 5.11(a)), la modélisation par morceaux permet d'obtenir une précision meilleure que celle du modèle cubique pour un coût de calcul légèrement plus faible ; le domaine initial a été découpé en 2 sous-domaines. En abaissant la valeur de $RMSE_{max}$ à 5 kHz (cf. Figure 5.11(b)), nous obtenons un modèle encore plus précis, mais le nombre de simulations pour le construire est plus grand ; le domaine initial a été découpé cette fois-ci en 8 sous-domaines.

Parmi les trois approches, la modélisation par morceaux est donc celle qui permet d'obtenir le modèle le plus précis. De plus, grâce à la valeur seuil $RMSE_{max}$, la modélisation par morceaux offre une flexibilité supplémentaire, que n'ont pas les deux autres approches, pour maîtriser la précision et le coût de calcul associé. Enfin, la valeur seuil $RMSE_{max}$ garantit une précision constante sur l'ensemble du domaine d'étude, et ce, quelle que soit sa largeur. Cependant, la modélisation par morceaux proposée conduit à des discontinuités aux frontières des sous-domaines. Ces discontinui-

tés rendent impossible l'utilisation de ces modèles par morceaux dans les algorithmes d'optimisation qui nécessitent des fonctions continues.

5.4 Conclusion

Dans ce chapitre, nous avons présenté une méthode de modélisation par morceaux des performances basée sur des modèles linéaires ou quadratiques. L'objectif de cette méthode est d'approximer les non-linéarités des performances sur des domaines d'étude relativement larges en prenant en compte les différents types de paramètres qui peuvent influencer les performances, à savoir les paramètres de conception, technologiques et environnementaux. Le but recherché n'est pas tant de modéliser extrêmement finement les performances des circuits au prix d'un coût calculatoire excessif, mais plutôt de saisir leur forme générale sur l'ensemble du domaine pour un coût de calcul réduit. Dans cette optique, les points forts de la méthode mise en place sont :

- la modélisation par morceaux qui permet de couvrir tout le domaine d'étude en le découpant de façon judicieuse en sous-domaines sur lesquels les performances sont moins non-linéaires et donc plus facilement approximables par des polynômes de degré peu élevé,
- la limitation du coût de calcul grâce à une modélisation locale séquentielle, au critère de bisection basé sur le gradient, à la réutilisation des simulations et à la valeur seuil $RMSE_{max}$ qui permet d'ajuster le coût calculatoire en fonction du niveau de précision demandé.

La contrainte principale que cherchent à satisfaire les méthodes de modélisation est généralement celle de la précision. Malheureusement, la construction d'un modèle infiniment précis s'accompagne invariablement d'un coût calculatoire tout aussi infini, lié aux irrégularités du phénomène à modéliser, à la complexité du modèle retenu ou encore au nombre de paramètres en jeu. Une solution souvent proposée consiste à mettre en place une modélisation en plusieurs étapes : l'idée est de construire à chaque étape un modèle dont le coût est minimum, mais dont la précision est suffisante pour apporter un certain nombre d'informations qui vont permettre de réduire considérablement la complexité du problème (en supprimant les paramètres non-influents ou en réduisant le domaine d'étude par exemple). Ainsi, à l'étape suivante, le problème étant moins complexe, un modèle plus précis pourra être construit pour un coût de calcul plus faible que celui obtenu sans avoir simplifié le problème.

Dans un tel processus de modélisation, l'approche par morceaux peut donc constituer une première étape destinée à apporter rapidement des premières informations utiles pour la suite : comportement global du circuit sur l'ensemble des valeurs des paramètres, identification des paramètres influents, recherche des zones où les performances sont optimales. Une fois cette étape

d'exploration spatiale effectuée, une modélisation plus fine et plus complexe peut alors être réalisée sur les régions d'intérêt.

Chapitre 6 : Conception robuste des circuits analogiques

Dans ce chapitre, nous allons présenter une méthode dont le but est de garantir la robustesse des circuits analogiques vis-à-vis des dispersions paramétriques. Cette méthode met en œuvre, dans un algorithme d'optimisation par intervalles, les méthodes de modélisation et d'estimation des performances pire-cas présentées dans les chapitres précédents. Nous traiterons d'abord la modélisation des performances pire-cas et la conception robuste de circuits analogiques, puis nous détaillerons l'algorithme d'optimisation par intervalles mis en place. Enfin, les résultats obtenus sur des circuits analogiques seront présentés dans la dernière partie. L'association de l'approximation de Cornish-Fisher et des algorithmes d'optimisation par intervalles se révèle efficace pour concevoir de façon automatisée et optimale des circuits analogiques robustes.

6.1 Présentation de la méthode de conception robuste

Dans le Chapitre 4, nous avons présenté une méthode qui permet d'analyser les variations des performances grâce aux plans d'expériences et au développement limité de Cornish-Fisher. Grâce à cette méthode, les valeurs pire-cas des performances peuvent être estimées pour un circuit donné, c'est-à-dire un jeu donné de valeurs pour les paramètres de conception. Ces valeurs extrêmes des performances permettent de juger de la robustesse du circuit par rapport aux spécifications sur les performances assignées par le concepteur. Dans le Chapitre 5, nous avons proposé une méthode de modélisation par morceaux des performances. Dans ce chapitre, nous allons appliquer ces méthodes à la conception robuste des circuits analogiques. La conception robuste va plus loin que l'analyse de variabilité puisqu'il s'agit de réutiliser les résultats de l'analyse de variabilité dans un processus d'optimisation dont l'objectif est de déterminer les valeurs des paramètres de conception pour lesquelles le circuit sera robuste aux dispersions paramétriques. Comme nous l'avons vu dans le Chapitre 3, pour qu'une méthode de conception robuste soit efficace, elle doit mettre en place une méthode d'analyse de la variabilité ainsi qu'un algorithme d'optimisation globale dont les coûts calculatoires sont aussi faibles que possible. La méthode de conception robuste que nous avons développée s'articule donc autour de deux points clés : la modélisation des performances pire-cas en fonction des paramètres de conception, puis l'optimisation de ces performances pire-cas pour rendre le circuit robuste (cf. Figure 6.1). La modélisation des performances pire-cas est réalisée en appliquant la méthode des plans d'expériences et l'approximation de Cornish-Fisher, tandis que l'optimisation

globale est effectuée avec un algorithme d'optimisation par intervalles qui intègre des techniques de réduction d'intervalles afin de réduire le coût calculatoire. Plusieurs travaux ont montré l'intérêt de l'arithmétique par intervalles (cf. Annexe C) pour la conception de circuits analogiques. En effet, elle peut être appliquée afin de déterminer le domaine de faisabilité correspondant à des spécifications données, permettant ainsi de réduire le domaine de conception [121, 122] ou encore de choisir parmi plusieurs topologies de circuits celle qui est la plus adaptée [121]. Par ailleurs, l'arithmétique affine, qui est une extension de l'arithmétique des intervalles permettant de prendre en compte les corrélations entre les paramètres, est utilisée dans [84] afin de réaliser des circuits analogiques robustes aux variations. Dans cette dernière approche, les performances pire-cas sont estimées à partir des bornes de la région de tolérance des variations paramétriques, sans tenir compte de leur distribution, ce qui ne permet pas de définir le rendement paramétrique associé aux performances pire-cas calculées. Dans son principe, l'approche [84] s'apparente donc aux analyses pire-cas (cf. §3.2.2) et souffre des mêmes problèmes de surestimation. A la différence de [84], dans la méthode présentée ci-dessous, la distribution des variations paramétriques est prise en compte dans l'estimation des performances pire-cas, permettant ainsi de définir un rendement paramétrique pour la conception robuste réalisée et éviter un surdimensionnement.

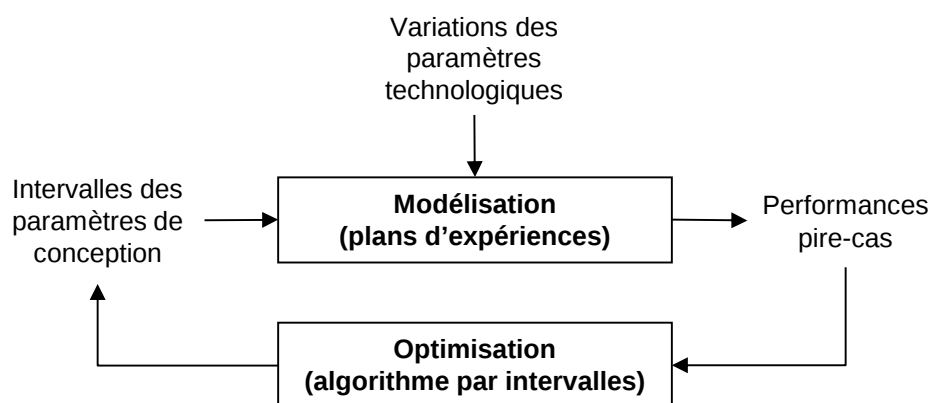


Figure 6.1 : Principe de la méthode de conception robuste

6.2 Conception robuste des circuits analogiques

Dans cette partie, nous allons présenter la méthode de modélisation des performances pire-cas en fonction des paramètres de conception, puis nous donnerons la formulation du problème d'optimisation dont la solution conduit à un dimensionnement robuste.

6.2.1 Robustesse d'un circuit

Un circuit est robuste aux dispersions paramétriques si son rendement de fabrication tend vers 100%. Le rendement est défini par les spécifications que le concepteur impose au circuit ; elles peuvent s'exprimer de façon simple sous la forme d'inégalités. Soit un circuit ayant une seule performance d'intérêt Y dont la densité de probabilité et la fonction de répartition sont respectivement définies par f_Y et F_Y . Si la contrainte sur cette performance est définie par l'inégalité $Y < \bar{B}$, le rendement de fabrication vis-à-vis de cette performance s'écrit alors :

$$\eta = P(Y < \bar{B}) = \int_{-\infty}^{\bar{B}} f_Y(y) dy = F_Y(\bar{B}) \quad (6.1)$$

En pratique un rendement de 100% ne sera jamais atteignable, on peut cependant se fixer un rendement minimum α suffisamment élevé, par exemple 99%. Le circuit sera alors considéré robuste si son rendement est supérieur à α , ce qui équivaut à :

$$\begin{aligned} \eta > \alpha &\Leftrightarrow F_Y(\bar{B}) > \alpha \\ &\Leftrightarrow \bar{B} > F_Y^{-1}(\alpha) \end{aligned} \quad (6.2)$$

où $F_Y^{-1}(\alpha)$ est le α -ième centile de Y , autrement dit la valeur pire-cas de Y correspondant à un rendement α . Par conséquent, pour garantir que le circuit sera robuste aux dispersions paramétriques, il suffit de comparer pour chaque performance ses valeurs pire-cas avec les spécifications (cf. Figure 6.2).

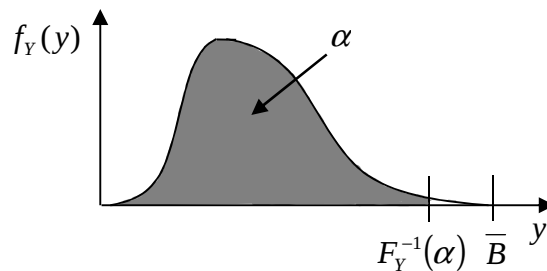


Figure 6.2 : Comparaison d'une spécification $\bar{B}$ et de la performance pire-cas $F_Y^{-1}(\alpha)$ pour un rendement minimum α donné

6.2.2 Modélisation des performances pire-cas

Nous avons vu dans le Chapitre 4 que les quantiles extrêmes d'une distribution pouvaient être estimés à partir de l'approximation de Cornish-Fisher :

$$\begin{aligned}\hat{Y}_{0.01} &\approx F_Y^{-1}(0.01) \\ \hat{Y}_{0.99} &\approx F_Y^{-1}(0.99)\end{aligned}\tag{6.3}$$

Cependant, les valeurs pire-cas $\hat{Y}_{0.01}$ et $\hat{Y}_{0.99}$ ainsi obtenues sont définies pour un jeu de valeurs données des paramètres de conception $\mathbf{C}_0$. L'objectif maintenant est donc de pouvoir prédire ces valeurs extrêmes pour différentes valeurs des paramètres de conception. Pour cela, nous avons mis en place une méthode de modélisation des performances pire-cas en fonction des paramètres de conception $\mathbf{C}$ qui aboutit au modèle suivant : $\hat{Y}_{extr} = \hat{f}_{extr}(\mathbf{C})$ où $\hat{Y}_{extr}$ représente indifféremment la borne inférieure $\hat{Y}_{0.01}$ ou la borne supérieure $\hat{Y}_{0.99}$ des variations des performances.

Notre méthode de modélisation des performances pire-cas est basée sur les plans d'expériences et fait appel à la construction séquentielle des modèles polynomiaux que nous avons déjà vue au Chapitre 5. Cette approche séquentielle consiste à essayer d'approximer la grandeur à modéliser d'abord par un modèle linéaire avec interactions grâce à un plan factoriel fractionnaire, puis si ce modèle n'est pas suffisant, par un modèle quadratique avec un plan composite centré. Les avantages de la modélisation séquentielle, notamment en termes de réutilisation des essais du plan factoriel, ont été détaillés dans la partie §5.2.2 du Chapitre 5.

La construction du modèle des performances pire-cas en fonction des paramètres de conception s'effectue en 2 étapes. La modélisation séquentielle par plans d'expériences est d'abord appliquée pour approximer les performances en fonction des paramètres de conception $\mathbf{C}$ et des variations technologiques $\Delta\mathbf{T}$. A chaque essai du plan d'expériences, une simulation circuit est réalisée avec un simulateur électrique comme Spice ou Eldo pour extraire les valeurs des performances. Le modèle obtenu, linéaire avec interactions ou quadratique, est noté $\hat{f}(\mathbf{C}, \Delta\mathbf{T})$. Si ce modèle n'est pas valide, le domaine d'étude est divisé en deux et la méthode de modélisation des performances pire-cas est appliquée aux deux sous-pavés. La modélisation séquentielle est ensuite appliquée une seconde fois pour approximer les performances pire-cas en fonction des paramètres de conception $\mathbf{C}$. A chaque essai du plan d'expériences, ce sont cette fois-ci les valeurs pire-cas des performances qui sont estimées à partir de la méthode que nous avons présentée au Chapitre 4 et qui est basée sur le modèle polynomial obtenu à l'étape précédente et l'approximation de Cornish-Fisher. Le flot de modélisation des performances pire-cas est illustré sur la Figure 6.3. Le modèle des performances pire-cas $\hat{f}_{extr}(\mathbf{C})$ est donc construit en appliquant deux fois la modélisation séquentielle. Cependant, le coût calculatoire de la deuxième modélisation séquentielle qui fait intervenir l'approximation de Cornish-Fisher est négligeable devant le coût de la première qui fait intervenir un simulateur électrique. Ce modèle des performances pire-cas va pouvoir être utilisé dans un algorithme

d'optimisation pour déterminer les valeurs des paramètres de conception C qui assurent un dimensionnement robuste.

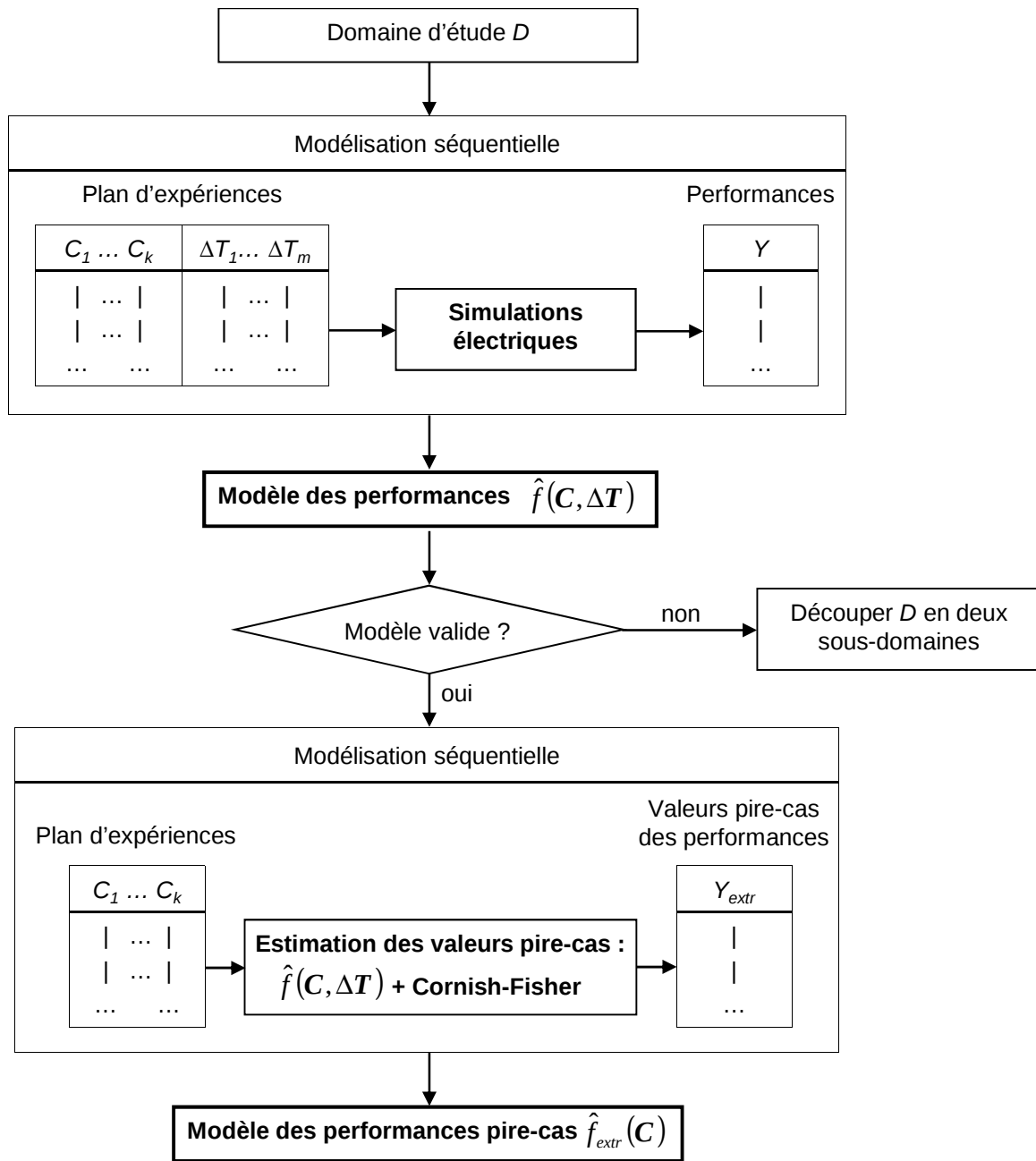


Figure 6.3 : Principe de la modélisation des performances pire-cas

6.2.3 Formulation sous la forme d'un problème d'optimisation robuste

Grâce au modèle des performances pire-cas $\hat{f}_{extr}(C)$, il est possible d'évaluer les valeurs extrêmes d'une performance, les comparer avec sa spécification et ainsi déterminer les valeurs des

paramètres de conception pour lesquelles la spécification est respectée. La conception robuste peut alors s'exprimer sous la forme d'un problème d'optimisation robuste tel que nous l'avons défini au Chapitre 3 (voir §3.3.4) :

$$\begin{array}{ll} \underset{\mathbf{C}}{\text{minimiser}} & \hat{f}_{\max}(\mathbf{C}) \\ \text{tel que} & \hat{g}_{l,\max}(\mathbf{C}) < 0, \quad l \in \{1, \dots, p\} \\ & \mathbf{C} \in D \end{array} \quad (6.4)$$

où

$\mathbf{C}$ est le vecteur des paramètres de conception qui appartient au domaine d'étude D ,

$\hat{f}_{\max}(\mathbf{C})$ est le modèle pire-cas de la performance à minimiser,

$\hat{g}_{l,\max}(\mathbf{C})$ est le modèle pire-cas des contraintes.

6.3 Algorithmes d'optimisation globale par intervalles

Dans la partie précédente, le problème de conception robuste a été formulé sous la forme d'un problème d'optimisation. Pour résoudre ce problème, plusieurs algorithmes d'optimisation sont envisageables. Nous allons, dans un premier temps, expliquer pourquoi notre choix s'est porté sur les algorithmes par intervalles, puis nous présenterons trois algorithmes d'optimisation par intervalles.

6.3.1 Optimisation globale avec contraintes

a) Formulation du problème

Dans la partie précédente, nous avons vu que la conception robuste des circuits analogiques passe par la résolution du problème d'optimisation (6.4). Il s'agit d'un problème d'optimisation globale avec contraintes. Dans la suite de cette partie, ce problème est exprimé sous la forme générale suivante :

$$\begin{array}{ll} \underset{\mathbf{X}}{\text{minimiser}} & f(\mathbf{X}) \\ \text{tel que} & g_l(\mathbf{X}) < 0, \quad l \in \{1, \dots, p\} \\ & \mathbf{X} \in D \end{array} \quad (6.5)$$

où $D \subset \mathbb{R}^k$, tandis que f et g_l sont des fonctions scalaires de $\mathbb{R}^k$ dans $\mathbb{R}$.

$$C = \{ \mathbf{X} \in D \mid g_l(\mathbf{X}) < 0, l \in \{1, \dots, p\} \} \quad (6.6)$$

est le domaine de faisabilité de (6.5). La difficulté des problèmes d'optimisation globale avec contraintes se situe à deux niveaux : déterminer les points qui appartiennent au domaine de faisabilité du problème et identifier leur minimum global.

b) Algorithmes de résolution

Les problèmes d'optimisation sont l'objet de nombreux travaux de recherche et les méthodes numériques permettant de les résoudre ne manquent pas. Le choix de la méthode de résolution la plus adaptée dépend de la nature du problème et des fonctions en jeu (cf. Annexe B). Dans le cadre de la conception robuste, les fonctions en jeu sont les performances pire-cas des circuits qui, comme nous l'avons vu, peuvent s'approximer par des modèles polynomiaux par morceaux. Il s'agit donc d'un problème d'optimisation avec contraintes, non-linéaire et non convexe : plusieurs optima locaux peuvent exister. Deux grandes familles de méthodes existent pour résoudre ce type de problème : les méthodes stochastiques (recuit-simulé, algorithme génétique, etc.) et les méthodes déterministes parmi lesquelles les algorithmes d'optimisation par intervalles. L'inconvénient majeur de toutes ces méthodes est leur coût en termes de temps de calcul et de ressources mémoires. Cependant, les algorithmes d'optimisation par intervalles présentent dans notre cas plusieurs avantages :

- la garantie d'obtenir l'optimum global du problème (que le domaine de faisabilité soit connexe ou non),
- les conditions sur la fonction de coût et les contraintes sont minimales (le calcul des dérivées n'est pas nécessaire),
- et surtout le raisonnement sur des intervalles, qui jouent un rôle clé dans ces algorithmes, se prête bien à la modélisation par morceaux des performances que nous avons développée dans le Chapitre 5.

6.3.2 Algorithme par intervalles de base

Les algorithmes d'optimisation que nous allons étudier s'appuient sur l'arithmétique des intervalles. Après avoir présenté succinctement les principes de cette arithmétique, nous détaillerons les différentes étapes des algorithmes d'optimisation par intervalles, puis nous donnerons leurs principales caractéristiques.

a) Arithmétique des intervalles

Les bases de l'arithmétique des intervalles ont été établies par R. Moore [123] dans les années 1960. Le principe de cette arithmétique est de ne pas manipuler des nombres, mais des intervalles qui les contiennent afin de garantir l'exactitude des résultats numériques. Elle fut donc initialement mise au point pour contrôler les erreurs de quantification causées par la représentation des nombres réels en nombres flottants. Plus récemment, l'arithmétique des intervalles fut étendue à l'optimisation et à la résolution de systèmes non-linéaires afin de fournir des résultats garantis [120, 124]. Les éléments de base de l'arithmétique des intervalles sont détaillés dans l'Annexe C.

b) Algorithmes par séparation et évaluation

Les algorithmes d'optimisation par intervalles appartiennent à la famille plus générale des algorithmes par séparation et évaluation ou « Branch and Bound » en anglais. En effet, ces algorithmes permettent de déterminer le minimum global grâce à deux étapes fondamentales :

- la séparation qui consiste à découper récursivement le problème en sous-problèmes,
- l'évaluation dont le but est d'éliminer les sous-problèmes en démontrant qu'ils ne satisfont pas toutes les contraintes du problème ou qu'ils ne conduisent pas à une solution meilleure que celle obtenue jusqu'à présent.

Les algorithmes par intervalles du type « Branch and Bound » font appel à l'arithmétique des intervalles, notamment aux fonctions d'inclusion des fonctions de coût et des contraintes.

c) Principe

Un algorithme par intervalles élémentaire pour résoudre le problème d'optimisation (6.5) est illustré sur la Figure 6.6. A chaque itération de la boucle principale, les sous-pavés contenant le minimum recherché sont décomposés en sous-pavés plus fins (étape de séparation) tandis que les sous-pavés ne contenant pas le minimum global ou ne satisfaisant pas les contraintes sont éliminés (étape d'évaluation). L'exécution de cet algorithme fournit une liste de pavés qui encadrent précisément le vecteur $\mathbf{X}^*$ solution du problème d'optimisation (6.5). Les différentes étapes de l'algorithme illustré sur la Figure 6.6 sont expliquées ci-dessous.

Initialisation

L'algorithme s'articule autour d'une liste L de couples $([\mathbf{X}], \underline{F})$ où $[\mathbf{X}]$ est un pavé et $\underline{F}$ la borne inférieure de l'image de F sur $[\mathbf{X}]$; cette borne inférieure est donc calculée avec la fonction d'inclusion naturelle F de f . Initialement, cette liste ne contient qu'un seul couple : $(D, \underline{F})$ où D est

le domaine d'étude des variables du problème d'optimisation (6.5) et $\underline{F}$ la borne inférieure de l'image de F sur D . Une autre liste R permet de sauvegarder les sous-pavés qui sont les solutions du problème d'optimisation (6.5) ; elle est initialement vide. Une variable réelle M est utilisée pour borner le minimum de f ; elle est initialisée à $+\infty$. Par ailleurs, une constante ε permet de fixer la largeur maximale des sous-pavés solutions contenus dans la liste R ; elle détermine la précision de l'encadrement du minimum. Enfin, une variable booléenne *toutes_les_contraintes_satisfaites* est utilisée pour étudier la satisfaction des contraintes.

Recherche du minimum

Afin de converger rapidement vers une solution, le sous-pavé $[X]$ qui est retiré de la liste L à chaque itération pour être traité en priorité est celui qui a la borne inférieure $\underline{F}$ la plus petite et qui est donc susceptible de contenir le minimum de la fonction de coût f sans tenir compte des contraintes (cf. Figure 6.4). Si la borne inférieure $\underline{F}$ est plus grande que la borne supérieure actuelle M sur le minimum f^* , alors le sous-pavé $[X]$ ne contient pas le minimum recherché ; le sous-pavé $[X]$ est donc supprimé et un nouveau sous-pavé est retiré de la liste L .

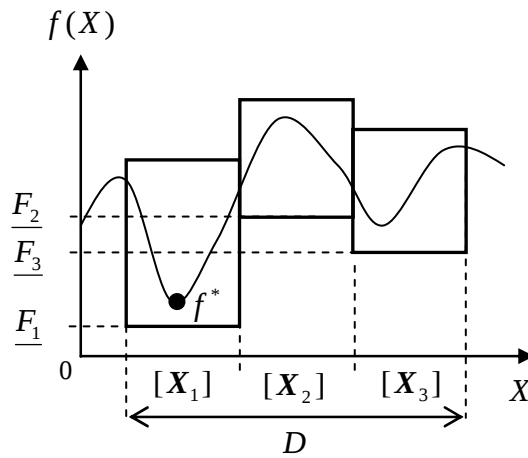


Figure 6.4 : Le sous-pavé qui est extrait de la liste L en priorité est celui qui présente la plus petite borne inférieure $\underline{F}$. Dans l'exemple ci-dessus, c'est donc le sous-pavé $[X_1]$ qui est extrait de L .

Etude de la satisfaction des contraintes

Les contraintes du problème d'optimisation (6.5) sont définies par les inégalités :

$$g_l(X) < 0, \quad l \in \{1, \dots, p\} \quad (6.7)$$

La satisfaction de chaque contrainte est vérifiée à partir des bornes $\underline{G}_l$ et $\overline{G}_l$ de l'image de G_l sur $[X]$ où G_l est la fonction d'inclusion naturelle de chaque fonction g_l . En effet, d'après la définition de la fonction d'inclusion (cf. Annexe C), on a la propriété suivante :

$$\forall X \in [X], \quad g_l(X) \in [\underline{G}_l, \overline{G}_l] \quad (6.8)$$

Ainsi, pour chaque contrainte, en comparant les valeurs de $\underline{G}_l$ et $\overline{G}_l$ par rapport à 0, trois cas de figure sont possibles :

- si $\underline{G}_l \geq 0$ alors aucun point du sous-pavé $[X]$ ne satisfait la contrainte (cf. Figure 6.5(a)), le sous-pavé est donc éliminé et un autre sous-pavé est retiré de la liste L ,
- si $\overline{G}_l < 0$ alors tous les points du sous-pavé $[X]$ satisfont la contrainte (cf. Figure 6.5(b)), le sous-pavé est donc conservé et la variable booléenne *toutes_les_contraintes_satisfaites* est instanciée avec la valeur (*toutes_les_contraintes_satisfaites* & Vrai) où & représente l'opération booléenne ET logique,
- si $\underline{G}_l < 0$ et $\overline{G}_l \geq 0$ alors certains points du sous-pavé $[X]$ satisfont la contrainte et d'autres non (cf. Figure 6.5), la contrainte est dite indéterminée, le sous-pavé est donc conservé et la variable *toutes_les_contraintes_satisfaites* est instanciée avec la valeur Faux.

Ces trois cas sont testés pour toutes les contraintes du problème d'optimisation. A l'issue de ces tests, si la variable *toutes_les_contraintes_satisfaites* est toujours égale à Vrai, alors le pavé $[X]$ est inclus dans le domaine de faisabilité du problème d'optimisation (toutes les contraintes sont satisfaites sur $[X]$). Dans le cas contraire, il existe des points de $[X]$ qui ne vérifient pas au moins une contrainte. L'objectif est donc de découper $[X]$ en sous-pavés plus fins jusqu'à ce qu'il n'y ait plus de contraintes indéterminées.

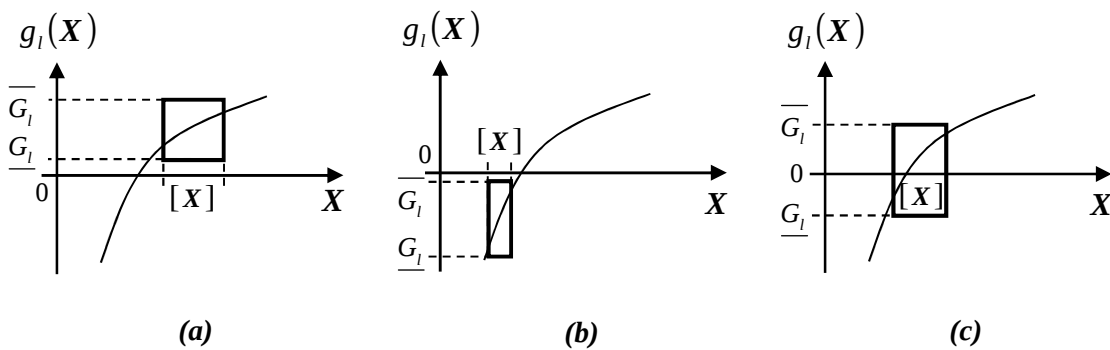


Figure 6.5 : Cas de figures possibles lors de l'étude de la satisfaction des contraintes à partir des bornes de l'image de G_l sur $[X]$

Mise à jour de la borne supérieure sur le minimum

Si toutes les contraintes sont satisfaites sur le sous-pavé $[X]$ qui a été extrait de la liste L , alors la borne supérieure M sur le minimum global f^* est mise à jour en lui assignant la plus petite valeur entre sa valeur actuelle et la borne supérieure $\bar{F}$ de F sur $[X]$.

Test de la largeur du sous-pavé

La largeur du sous-pavé $[X]$ est ensuite testée afin de décider de sa bisection ou non. Si sa largeur est inférieure à la constante ε et que toutes les contraintes sont satisfaites, alors $[X]$ encadre finement la solution du problème (6.5). Si la largeur du sous-pavé $[X]$ est inférieure à ε , mais que toutes les contraintes ne sont pas satisfaites, alors $[X]$ est supprimé de la liste L .

Bisection du sous-pavé

Dans le cas où la largeur du sous-pavé $[X]$ est supérieure à ε , il est décomposé en deux sous-pavés $[X_1]$ et $[X_2]$ sur lesquels la borne inférieure de l'image de F est calculée. Ces deux sous-pavés sont ensuite ajoutés à la liste L et une nouvelle itération de l'algorithme est effectuée. La façon selon laquelle la bisection est réalisée joue un rôle important dans la rapidité de convergence de l'algorithme [125]. Plusieurs méthodes de bisection sont envisageables :

- la décomposition la plus utilisée consiste à bissecter l'intervalle le plus large de $[X]$, c'est-à-dire la direction i qui maximise :

$$(\bar{X}_i - \underline{X}_i), \quad i = 1 \dots k \quad (6.9)$$

La décomposition est donc réalisée de façon uniforme.

- Si les variables d'optimisation présentent des ordres de grandeur très différents, mais sont toutes positives (ce qui est généralement le cas des grandeurs physiques), la décomposition peut alors s'effectuer perpendiculairement à la direction i qui maximise :

$$\frac{(\bar{X}_i - \underline{X}_i)}{(\bar{X}_i + \underline{X}_i)/2}, \quad i = 1 \dots k \quad (6.10)$$

- Enfin, le critère de bisection, exposé dans le Chapitre 5 et qui consiste à bissecter la direction pour laquelle la variation du gradient est maximale, peut également être appliqué (cf. §5.2.4).

Mise à jour de la liste de résultat

Enfin, lorsqu'il n'y a plus de sous-pavés à traiter (la liste L est vide) et si une solution existe, la liste R est non vide et contient les sous-pavés qui sont solutions du problème d'optimisation (6.5). Les sous-pavés dont la borne inférieure $\underline{F}$ est supérieure à M sont éliminés de la liste R . Lorsque l'algorithme se termine, la liste R contient donc des sous-pavés dont l'union encadre précisément le vecteur X^* solution de (6.5).

I) Initialisation
1. $L = \{(D, \overline{F})\}$, $R = \emptyset$, $M = +\infty$, ε , <i>toutes_les_contraintes_satisfaites</i> = Vrai
II) Recherche du minimum
2. Tant que la liste L n'est pas vide Faire 3. Retirer de L le sous-pavé $[X]$ qui a la plus petite borne inférieure $\underline{F}$ 4. Si $\underline{F} > M$ Alors 5. Supprimer $[X]$ de L 6. Aller à l'étape 2 7. Fin Si
III) Etude de la satisfaction des contraintes
8. Pour l allant de 1 à p Faire 9. Calculer l'image de G_l sur $[X]$ avec l'arithmétique des intervalles : $[\underline{G}_l, \overline{G}_l] = G_l([X])$ 10. Si $\underline{G}_l \geq 0$ Alors 11. Supprimer $[X]$ de L 12. Aller à l'étape 2 13. Sinon Si $\overline{G}_l < 0$ Alors 14. <i>toutes_les_contraintes_satisfaites</i> = <i>toutes_les_contraintes_satisfaites</i> & Vrai 15. Sinon 16. <i>toutes_les_contraintes_satisfaites</i> = Faux 17. Fin si 18. Fin Si 19. Fin Pour
IV) Mise à jour de la borne supérieure sur le minimum
20. Si <i>toutes_les_contraintes_satisfaites</i> = Vrai Alors 21. Calculer la borne supérieure $\overline{F}$ de F sur $[X]$ 22. $M = \min(M, \overline{F})$ 23. Fin Si
V) Test de la largeur du sous-pavé

24.	Si $w([X]) < \varepsilon$ Alors
25.	Si <i>toutes_les_contraintes_satisfaites</i> = Vrai Alors
26.	Ajouter $[X]$ à la liste R
27.	Sinon
28.	Supprimer $[X]$ de L
29.	Fin Si
VI) Bissection du sous-pavé	
30.	Sinon
31.	Décomposer $[X]$ en deux sous-pavés $[X_1]$ et $[X_2]$
32.	Calculer les bornes inférieures $\underline{F}_1$ et $\underline{F}_2$ de F sur $[X_1]$ et $[X_2]$
33.	Ajouter $([X_1], \underline{F}_1)$ et $([X_2], \underline{F}_2)$ à L
34.	Fin Si
35.	Fin Tant que
VII) Mise à jour de la liste de résultat	
36.	Si la liste R n'est pas vide Alors
37.	Supprimer de la liste R les sous-pavés pour lesquels $\underline{F} > M$
38.	Fin Si
39.	Retourner la liste R

Figure 6.6 : Algorithme d'optimisation par intervalles

d) Convergence de l'algorithme

La terminaison de l'algorithme est assurée. En effet, la décomposition du domaine d'étude initial lors de l'exécution de l'algorithme conduit à des sous-pavés de plus en plus fins qui, à partir d'un certain seuil de précision fixé par la constante ε , seront ajoutés à la liste R (s'ils satisfont les contraintes et améliorent le minimum déjà trouvé) ou bien supprimés dans le cas contraire. Ainsi, suivant la valeur de la constante ε , la convergence de l'algorithme sera plus ou moins rapide.

e) Conclusion

Le principal avantage de l'algorithme par intervalles présenté ci-dessus est de fournir le minimum global du problème d'optimisation (6.5). A cela s'ajoute le fait que les conditions sur les fonctions du problème d'optimisation ne sont pas contraignantes : le gradient des fonctions n'est pas nécessaire et leur fonction d'inclusion naturelle peut être utilisée pour les calculs avec l'arithmétique des intervalles.

Cependant, l'inconvénient majeur de cet algorithme réside dans la procédure récursive de bissection : à chaque itération, les sous-pavés sont découpés en deux ce qui conduit à une augmentation exponentielle de la complexité du problème avec le nombre de paramètres. Ainsi, le nombre de

sous-pavés en attente de traitement dans la liste L croît rapidement lorsque le nombre de paramètres est élevé et que le seuil ε sur la largeur des sous-pavés de liste R est faible. Afin d'accélérer la convergence de l'algorithme par intervalles, des méthodes de réduction des sous-pavés ont donc été développées.

6.3.3 Algorithmes par intervalles avec contracteurs

Nous venons de voir que chaque bisection ralentit la convergence de l'algorithme. Afin d'accélérer la recherche de l'optimum, la bisection des pavés ne doit donc être utilisée qu'en dernier recours. Une technique classique d'accélération des algorithmes par intervalles consiste ainsi à réduire autant que possible la taille des pavés avant de les découper. En effet, à partir de l'analyse par intervalles, il est possible de répercuter sur les pavés un certain nombre d'informations extraites des contraintes. Grâce à ces informations, des intervalles de points qui n'appartiennent pas au domaine de faisabilité du problème peuvent alors être éliminés. Les algorithmes qui réduisent la taille des pavés sont appelés des contracteurs. L'intérêt principal des contracteurs réside dans leur complexité polynomiale à comparer avec la complexité exponentielle de la procédure de bisection. Après avoir donné une définition plus précise des contracteurs, nous présenterons deux d'entre eux. Leur intégration dans un algorithme d'optimisation sera ensuite détaillée.

a) Problème de satisfaction de contraintes

Formulation du problème

La réduction des pavés (« pruning » en anglais), en éliminant les points qui ne satisfont pas aux contraintes du problème d'optimisation (6.5), revient donc à déterminer le domaine de faisabilité (6.6) de ce dernier. Le domaine de faisabilité peut s'exprimer sous la forme d'un problème de satisfaction de contraintes (CSP - « Constraint Satisfaction Problem » en anglais). Un problème de satisfaction de contraintes se compose :

- d'un vecteur de variables $\mathbf{X} = (X_1, X_2, \dots, X_k)^T \in \mathfrak{R}^k$ et d'un vecteur contenant les valeurs des contraintes $\mathbf{Y} = (Y_1, Y_2, \dots, Y_p)^T \in \mathfrak{R}^p$,
- des pavés correspondants $[\mathbf{X}] = [\underline{X}_1, \overline{X}_1] \times [\underline{X}_2, \overline{X}_2] \times \dots \times [\underline{X}_k, \overline{X}_k]$ qui définit le domaine d'étude de $\mathbf{X}$ et $[\mathbf{Y}] = [\underline{Y}_1, \overline{Y}_1] \times [\underline{Y}_2, \overline{Y}_2] \times \dots \times [\underline{Y}_p, \overline{Y}_p]$ qui définit le domaine de $\mathbf{Y}$,
- d'un ensemble de contraintes exprimées par les relations $g_l(\mathbf{X}) = Y_l, l \in \{1, \dots, p\}$.

Le problème de satisfaction de contraintes peut s'exprimer sous la forme vectorielle suivante :

$$CSP : (g(X) = Y, X \in [X], Y \in [Y]) \quad (6.11)$$

où $g(X) = (g_1(X), g_2(X), \dots, g_p(X))^T$. L'ensemble des solutions de (6.11) s'écrit :

$$S = \{X \in [X] \mid g(X) \in [Y]\} \quad (6.12)$$

Les contraintes d'un problème *CSP* peuvent aussi bien s'exprimer sous la forme d'égalités que d'inégalités. Par exemple, le système de contraintes

$$\begin{cases} X_1 + X_2 \leq 0 \\ X_1 - X_2 = 4 \\ X_1 \in \Re, X_2 \in \Re \end{cases} \quad (6.13)$$

peut se formuler sous la forme d'un problème *CSP* en introduisant deux variables supplémentaires Y_1 et Y_2 afin d'obtenir le système suivant :

$$\begin{cases} X_1 + X_2 = Y_1 \\ X_1 - X_2 = Y_2 \\ X_1 \in \Re, X_2 \in \Re, \\ Y_1 \in]-\infty, 0], Y_2 \in [4, 4] \end{cases} \quad (6.14)$$

On retrouve alors la forme d'un problème *CSP* (6.11) en posant $g(X) = (X_1 + X_2, X_1 - X_2)^T$, $[X] =]-\infty, +\infty[\times]-\infty, +\infty[$ et $[Y] =]-\infty, 0] \times [4, 4]$.

Ainsi, déterminer le domaine de faisabilité (6.6) d'un problème d'optimisation revient à déterminer l'ensemble S des solutions du problème *CSP* (6.11).

Contracteurs

Une méthode efficace pour résoudre le problème *CSP* (6.11) consiste à déterminer le plus petit sous-pavé qui contient l'ensemble S des solutions : c'est le rôle des contracteurs. Un contracteur pour le problème de satisfaction de contraintes (6.11) se définit donc comme un opérateur C_{CSP} qui génère, à partir d'un pavé $[X]$, un pavé $[X'] = C_{CSP}([X])$ vérifiant les propriétés suivantes [120] :

$$\forall [X] \in \mathbb{R}^k, [X'] \subset [X] \quad (6.15)$$

$$\forall [X] \in \mathbb{R}^k, [X] \cap S \subset [X'] \quad (6.16)$$

La propriété (6.15) signifie que le nouveau pavé $[X']$ est inclus dans le pavé initial $[X]$ (contraction), tandis que d'après la propriété (6.16), tous les points qui appartiennent à l'ensemble S des solutions sont contenus dans le pavé contracté $[X']$ (aucun point solution n'a été écarté).

Plusieurs contracteurs ont été développés en étendant aux intervalles les algorithmes classiques de résolution des systèmes d'équations linéaires ou non-linéaires comme les contracteurs de Gauss, de Newton ou de Krawczyk [120, 126]. Cependant, ces contracteurs ne sont applicables que si le nombre de contraintes est égal au nombre de variables. Dans la suite, nous allons donc présenter deux contracteurs qui s'appliquent quel que soient le nombre de contraintes et le nombre de variables. Le premier est basé sur des techniques de propagation de contraintes, tandis que le second repose sur la résolution de problèmes d'optimisation quadratique avec contraintes.

b) Contracteur par propagation – rétropropagation

Ce contracteur est issu de l'adaptation aux intervalles des techniques de propagation de contraintes utilisées en programmation logique [120, 127]. Le cœur de cet algorithme repose sur une procédure de propagation-rétropropagation appliquée séparément à chaque contrainte du problème *CSP* afin de contracter le domaine de chaque variable. Cette procédure est répétée jusqu'à ce que la contraction ne soit plus efficace.

Arbre binaire de représentation des contraintes

Les fonctions g_i des contraintes du problème (6.11) sont d'abord décomposées en une séquence d'opérations et de fonctions élémentaires (addition, soustraction, multiplication, division, sinus, cosinus, logarithme, etc...) dont la fonction réciproque est connue. Elles sont ensuite représentées sous la forme d'un arbre binaire :

- chaque feuille représente une variable de la contrainte et son domaine associé ou bien une constante,
- chaque nœud correspond à la fonction d'inclusion d'une opération ou d'une fonction élémentaire.

Procédure de propagation-rétropropagation

Etant donnée une contrainte $g_i(X) = Y_i$, la contraction du domaine $[X]$ des variables de cette contrainte est effectuée en deux phases successives :

1. Une phase dite de propagation où l'arbre est traversé depuis ses feuilles jusqu'à sa racine. A chaque nœud, la fonction d'inclusion de l'opération correspondante est appliquée sur les intervalles des nœuds fils afin de calculer le résultat partiel au niveau du nœud étudié. Les domaines des variables sont ainsi propagés à travers les opérations élémentaires de l'arbre jusqu'à la ra-

cine ; le résultat obtenu au niveau de la racine correspond à l'intervalle image $G_l([X])$ de la fonction d'inclusion G_l de g_l sur le pavé initial $[X]$ des variables.

2. Une phase dite de rétropropagation où l'arbre est cette fois-ci traversé depuis sa racine jusqu'à ses feuilles. L'intervalle image de la racine est d'abord mis à jour en le remplaçant par son intersection avec l'intervalle de Y_l . Au cours de la traversée de l'arbre, on essaie ensuite de réduire l'intervalle partiel de chaque nœud (obtenu lors de la phase de propagation) en lui appliquant un opérateur de projection. L'objectif de la projection est de déterminer le plus petit intervalle encadrant l'ensemble solution ; la construction de cet opérateur est expliquée dans la partie suivante. En procédant ainsi de nœud en nœud, on parvient à réduire jusqu'aux intervalles des feuilles de l'arbre qui représentent les domaines des variables.

A l'issue de ces deux phases, le pavé initial des variables a été réduit dans le meilleur des cas ou reste inchangé dans le pire des cas. Lors de la phase de rétropropagation, si l'intervalle obtenu pour un nœud donné est vide, alors le problème CSP n'a pas de solution.

Projection de contraintes

Le cas simple d'une contrainte $x_g \bullet x_d = y$ qui ne fait intervenir que deux variables va nous permettre d'expliquer la construction des opérateurs de projection (elle se généralise pour chaque nœud de l'arbre). Les intervalles initiaux des variables x_g et x_d sont respectivement notés $[x_g]_0$ et $[x_d]_0$, tandis que celui de la contrainte est noté $[y]_0$. La contrainte étudiée n'implique qu'une seule opération élémentaire $\bullet$ dont la fonction d'inclusion est notée $\bullet$. Les fonctions réciproques à gauche et à droite de $\bullet$ sont respectivement notées $\bullet_g^{-1}$ et $\bullet_d^{-1}$. Par exemple, si $\bullet$ est l'addition, alors $\bullet^{-1}$ est la soustraction. L'ensemble des points qui vérifient la contrainte $x_g \bullet x_d = y$ est défini par :

$$S_\bullet = \{(x_g, x_d) \in [x_g] \times [x_d] \mid x_g \bullet x_d \in [y]\} \quad (6.17)$$

La projection de $S_\bullet$ sur le domaine x_g (respectivement x_d) est alors définie par l'expression (6.18) (respectivement (6.19)). Le principe de ces projections est illustré sur la Figure 6.7.

$$\Pi_g = \{x_g \in [x_g] \mid \exists x_d \in [x_d], x_g \bullet x_d \in [y]\} \quad (6.18)$$

$$\Pi_d = \{x_d \in [x_d] \mid \exists x_g \in [x_g], x_g \bullet x_d \in [y]\} \quad (6.19)$$

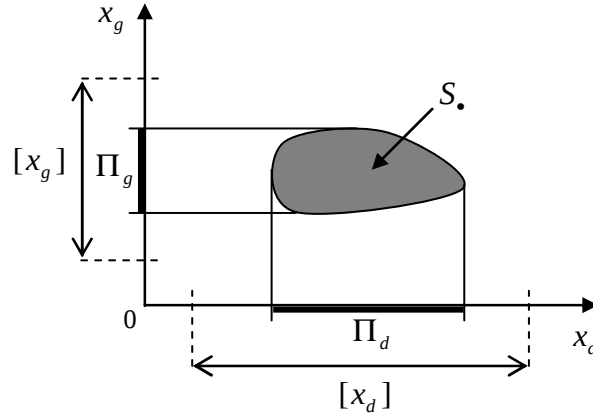


Figure 6.7 : Projections de l'ensemble solution $S_$ sur les domaines de x_g et x_d

Les projections de la contrainte sur les domaines de x_g et x_d permettent donc de déterminer le plus petit pavé contenant $S_$. La construction des opérateurs permettant de réaliser ces projections nécessite de reformuler la contrainte de façon à isoler les variables x_g et x_d :

$$\begin{aligned}
 x_g \bullet x_d = y &\Leftrightarrow x_g = y \bullet_d^{-1} x_d \\
 &\Leftrightarrow x_d = x_g \bullet_g^{-1} y
 \end{aligned}
 \tag{6.20}$$

A partir des expressions précédentes et en raisonnant sur des intervalles, on en déduit les expressions des opérateurs de projection :

$$\begin{aligned}
 x_g = y \bullet_d^{-1} x_d &\Rightarrow x_g \in [x_g]_0 \cap ([y] \bullet_d^{-1} [x_d]) \\
 x_d = x_g \bullet_g^{-1} y &\Rightarrow x_d \in [x_d]_0 \cap ([x_g] \bullet_g^{-1} [y])
 \end{aligned}
 \tag{6.21}$$

Les domaines des variables x_g et x_d sont donc contractés en appliquant respectivement les opérateurs de projection $[x_g]_0 \cap ([y] \bullet_d^{-1} [x_d])$ et $[x_d]_0 \cap ([x_g] \bullet_g^{-1} [y])$. L'algorithme de la procédure de propagation-rétropropagation appliquée à la contrainte $x_g \bullet x_d = y$ est représenté sur la Figure 6.8.

I) Propagation
1. $[y] = [x_g]_0 \bullet [x_d]_0$
II) Rétropropagation
2. Si $[y] = \emptyset$ Alors
3. le problème CSP n'a pas de solution
4. Sinon
5. $[x_g] = [x_g]_0 \cap ([y] \bullet_d^{-1} [x_d])$
6. $[x_d] = [x_d]_0 \cap ([x_g] \bullet_g^{-1} [y])$
7. Fin Si
8. Retourner $[x_g]$ et $[x_d]$

Figure 6.8 : Procédure de propagation-rétropropagation appliquée à la contrainte $x_g \bullet x_d = y$

Exemple de procédure de propagation-rétropropagation

Afin d'illustrer graphiquement la procédure de contraction, les modifications apportées sur l'arbre binaire associé à la contrainte $x_1.x_2 + 2x_3 = y$ sont représentées sur la Figure 6.9 (phase de propagation) et la Figure 6.10 (phase de rétropropagation). Les intervalles initiaux des variables sont $[x_1] = [1, 5]$, $[x_2] = [1, 5]$, $[x_3] = [-7, -10]$ et celui de la contrainte est $[y] = [0, 5]$. Après la contraction, les intervalles obtenus sont $[x_1] = [2.8, 5]$, $[x_2] = [2.8, 5]$ et $[x_3] = [-7, -10]$.

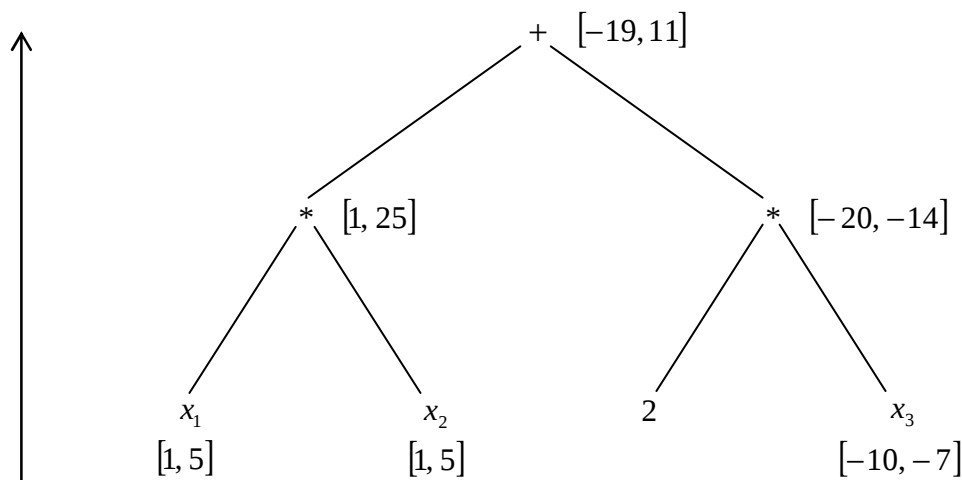


Figure 6.9 : Phase de propagation appliquée à la contrainte $x_1.x_2 + 2x_3 = y$

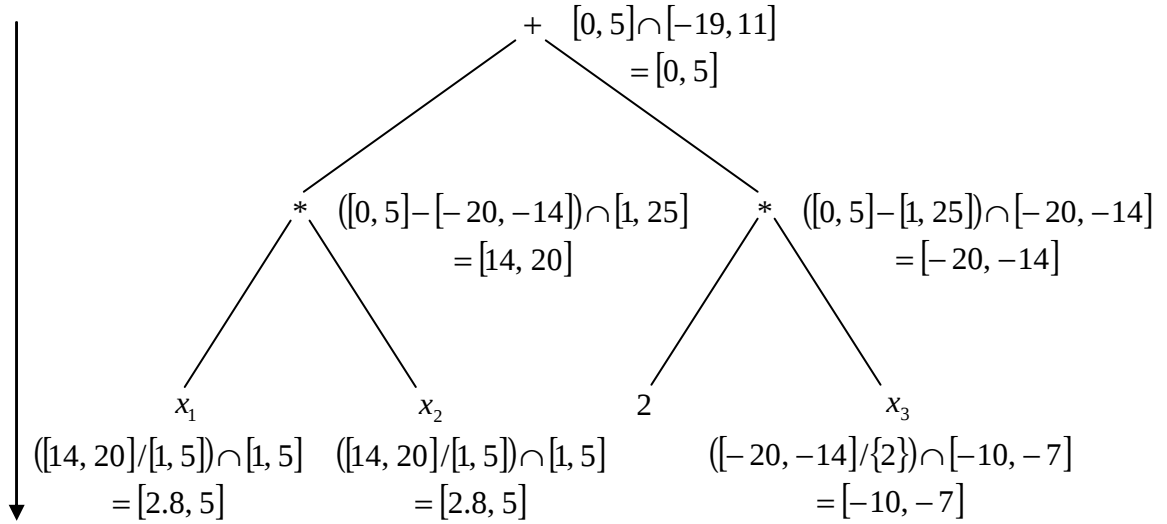


Figure 6.10 : Phase de rétropropagation appliquée à la contrainte $x_1 \cdot x_2 + 2x_3 = y$

Algorithme du contracteur par propagation-rétropropagation

La procédure de propagation-rétropropagation que nous venons de voir ne s'applique que sur une seule contrainte. Le contracteur par propagation-rétropropagation consiste donc à appliquer cette procédure sur chaque contrainte tant qu'une réduction est apportée au domaine des variables. L'algorithme de ce contracteur est illustré sur la Figure 6.11. L'ordre dans lequel les contraintes sont traitées n'a pas d'importance sur le pavé obtenu à la fin de la contraction. Par ailleurs, au fur et à mesure des itérations de la boucle principale, le nouveau pavé $[X']$ est de moins en moins réduit par rapport au pavé $[X]$. Aussi, une procédure (qui n'apparaît pas sur la Figure 6.11) est mise en place pour arrêter l'algorithme dès que la contraction des pavés n'est plus significative. Les contracteurs basés sur la propagation de contraintes sont particulièrement efficaces lorsque les domaines des variables sont larges [120]. Le logiciel libre « Interval Peeler » [128] implémente un contracteur de ce type et permet ainsi de tester la réduction d'intervalles. Par la suite, le contracteur par propagation-rétropropagation sera noté C_{PR} .

I) Initialisation
1. Pavé initial $[X]$, E = ensemble de contraintes $\{g_1(X) = Y_1, g_2(X) = Y_2, \dots, g_p(X) = Y_p\}$
II) Contraction du pavé
2. Tant que E n'est pas vide Faire 3. Choisir une contrainte $g_l(X) = Y_l$ de E 4. $[X'] \leftarrow$ procédure de propagation-rétropropagation appliquée à $([X], g_l(X) = Y_l)$ 5. Si $[X'] = \emptyset$ Alors 6. Fin de l'algorithme (le problème CSP n'a pas de solution) 7. Fin Si 8. Si $[X] \neq [X']$ Alors 9. $[X] \leftarrow [X']$ 10. Ajouter à E toutes les contraintes impliquant des variables dont le domaine a été réduit 11. Sinon 12. Supprimer de E la contrainte $g_l(X) = Y_l$ 13. Fin Si 10. Fin Tant que 11. Retourner $[X]$

Figure 6.11 : Algorithme du contracteur de propagation-rétropropagation

c) Contracteur par programmation quadratique

Dans [120], un contracteur s'appuyant sur un algorithme de programmation linéaire est présenté. Ce contracteur s'applique dans le cas particulier où les contraintes du problème CSP sont des fonctions linéaires. Dans la méthode de conception robuste que nous avons proposée, les performances des circuits sont approximées par des modèles quadratiques (les modèles linéaires avec interactions sont des cas particuliers des modèles quadratiques). Le problème de satisfaction des contraintes qui intervient dans le problème d'optimisation est donc :

$$CSP_{quad} : (g_{quad}(X) = Y, X \in [X], Y \in [Y]) \quad (6.22)$$

où $g_{quad}(X) = (g_{quad,1}(X), g_{quad,2}(X), \dots, g_{quad,p}(X))^T$ est un vecteur de fonctions quadratiques.

Nous avons donc développé un contracteur qui s'inspire du contracteur par programmation linéaire de [120], mais qui s'appuie sur un algorithme d'optimisation quadratique afin d'être compatible avec les fonctions quadratiques du problème CSP_{quad} .

L'ensemble des solutions de (6.22) est défini par :

$$\begin{aligned}
S_{quad} &= \{ \mathbf{X} \in [\mathbf{X}] \mid \mathbf{g}_{quad}(\mathbf{X}) \in [\mathbf{Y}] \} \\
\Leftrightarrow S_{quad} &= \{ \mathbf{X} \in [\mathbf{X}] \mid \mathbf{g}_{quad}(\mathbf{X}) \geq \underline{\mathbf{Y}} \text{ et } \mathbf{g}_{quad}(\mathbf{X}) \leq \overline{\mathbf{Y}} \}
\end{aligned} \tag{6.23}$$

où $[\mathbf{Y}] = [\underline{Y}_1, \overline{Y}_1] \times [\underline{Y}_2, \overline{Y}_2] \times \dots \times [\underline{Y}_p, \overline{Y}_p]$, $\underline{\mathbf{Y}} = (\underline{Y}_1, \underline{Y}_2, \dots, \underline{Y}_p)$ et $\overline{\mathbf{Y}} = (\overline{Y}_1, \overline{Y}_2, \dots, \overline{Y}_p)$. Le plus petit pavé encadrant l'ensemble solution (6.23) est construit en traitant les variables X_i du vecteur $\mathbf{X}$ une par une. Pour chaque variable, l'objectif est de déterminer le plus petit intervalle de valeurs pour lesquelles les contraintes du problème (6.22) sont respectées. Les bornes d'un tel intervalle peuvent donc être obtenues en résolvant les deux problèmes d'optimisation suivants :

$$\begin{aligned}
INF \quad & \begin{cases} \text{minimiser} & X_i \\ \mathbf{X} \\ \text{tel que} & \mathbf{g}_{quad}(\mathbf{X}) \geq \underline{\mathbf{Y}} \\ & \mathbf{g}_{quad}(\mathbf{X}) \leq \overline{\mathbf{Y}} \\ & \mathbf{X} \in [\mathbf{X}] \end{cases} \quad \quad \quad SUP \quad \begin{cases} \text{maximiser} & X_i \\ \mathbf{X} \\ \text{tel que} & \mathbf{g}_{quad}(\mathbf{X}) \geq \underline{\mathbf{Y}} \\ & \mathbf{g}_{quad}(\mathbf{X}) \leq \overline{\mathbf{Y}} \\ & \mathbf{X} \in [\mathbf{X}] \end{cases}
\end{aligned} \tag{6.24}$$

La valeur minimum de X_i obtenue en résolvant le problème *INF* donne la borne inférieure de l'intervalle, tandis que la valeur maximum de X_i obtenue en résolvant le problème *SUP* donne sa borne supérieure. Le plus petit pavé encadrant l'ensemble solution (6.23) est construit en effectuant ces deux optimisations pour chaque variable X_i du vecteur $\mathbf{X}$ pour i allant de 1 à k . Les fonctions qui interviennent dans les contraintes sont des fonctions quadratiques, il s'agit donc de problèmes d'optimisation avec contraintes quadratiques. Ce type de problème, généralement non-convexe, peut être efficacement résolu en ayant recours à une relaxation lagrangienne ou semidéfinie [129, 130]. Par la suite, le contracteur par programmation quadratique sera noté C_{PQ} .

d) Comparaison des contracteurs

Il est intéressant de comparer les deux contracteurs en termes de temps de calcul et de contraction. A première vue, le contracteur par programmation quadratique C_{PQ} , qui nécessite de résoudre $2k$ problèmes d'optimisations où k est le nombre de variables, semble plus lent que le contracteur par propagation-rétropropagation C_{PR} . Cependant, ce dernier fait appel à l'arithmétique des intervalles dont l'inconvénient majeur est la surestimation des encadrements fournis (cf. Annexe C), ce qui se traduit par des pavés moins contractés que ceux obtenus avec le contracteur par programmation quadratique. L'exemple suivant va permettre d'illustrer ces deux remarques. Soit le problème *CSP* :

$$\begin{cases} 4 \leq X_1 - 1.2X_2 + 0.2X_1X_2 \\ X_1^2 + X_2^2 \leq 80 \\ X_1 \in [0, 10], X_2 \in [0, 10] \end{cases}$$

$$\Leftrightarrow \begin{cases} X_1 - 1.2X_2 + 0.2X_1X_2 = Y_1 \\ X_1^2 + X_2^2 = Y_2 \\ X_1 \in [0, 10], X_2 \in [0, 10], \\ Y_1 \in [4, +\infty[, Y_2 \in]-\infty, 80] \end{cases} \quad (6.25)$$

Les contraintes $X_1 - 1.2X_2 + 0.2X_1X_2 = Y_1$ et $X_1^2 + X_2^2 = Y_2$ sont représentées graphiquement sur la Figure 6.12(a) et la Figure 6.12(b). Les deux contracteurs ont été codés sous Matlab : le contracteur C_{PR} implémente la toolbox INTLAB [131] pour effectuer les calculs sur les intervalles, tandis que le contracteur C_{PQ} utilise la fonction Matlab « *fmincon* » pour résoudre les problèmes d'optimisation. Les plus petits pavés, qui encadrent l'ensemble solution du problème (6.25) et qui sont générés avec chacun des contracteurs, sont donnés dans le Tableau 6.1 et représentés sur la Figure 6.13. Les résultats obtenus confirment nos deux hypothèses : la contraction réalisée avec le contracteur C_{PQ} est bien meilleure que celle effectuée avec le contracteur C_{PR} , mais son coût calculatoire est plus élevé.

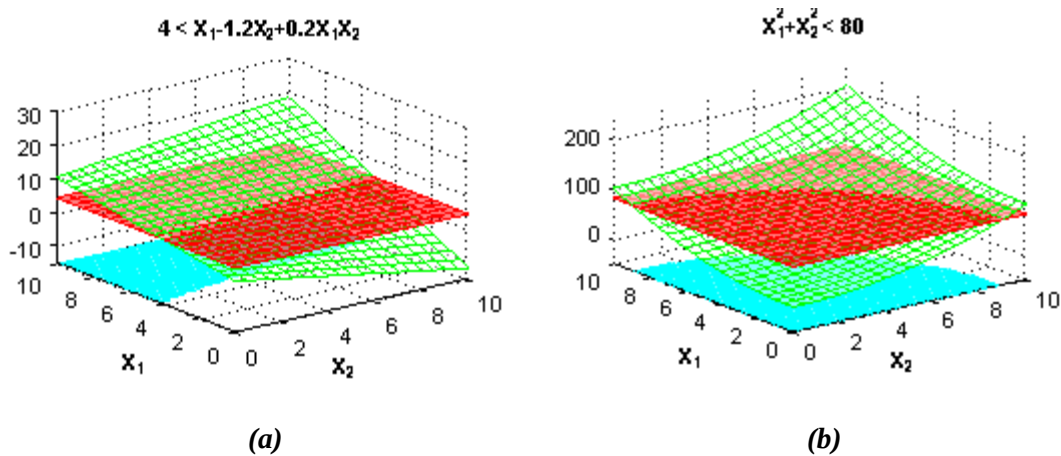


Figure 6.12 : Représentation graphique des contraintes du problème CSP (6.25). L'ensemble solution de chaque contrainte est représenté en bleu clair.

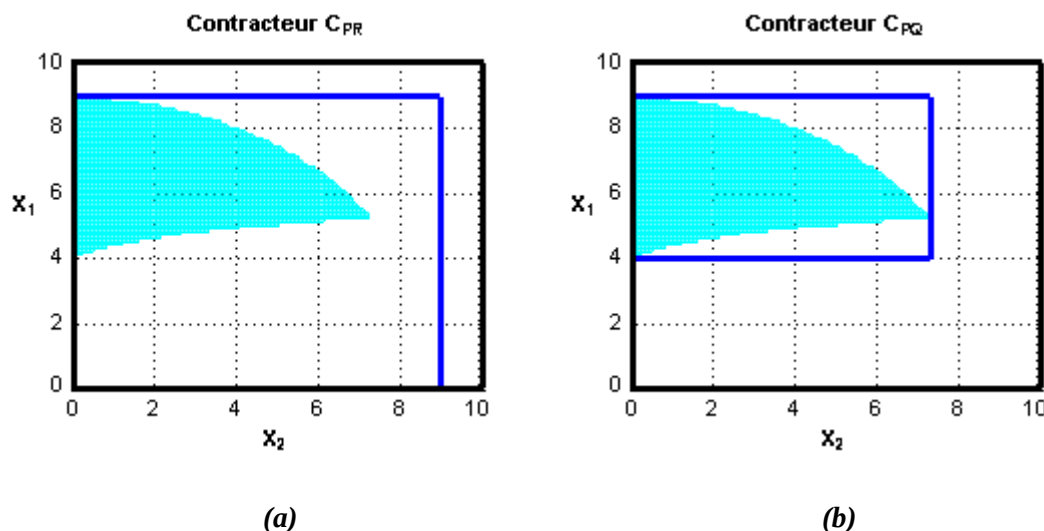


Figure 6.13 : Les pavés obtenus avec le contracteur par propagation-rétropropagation (a) et le contracteur par programmation quadratique (b) sont modélisés par les rectangles en bleu foncé. L'ensemble solution du problème CSP (6.25) est représenté en bleu clair.

Tableau 6.1 : Comparaison des contracteurs C_{PR} et C_{PQ}

	Contracteur C_{PR}	Contracteur C_{PQ}
Pavé initial	[0, 10][0, 10]	
Pavé contracté	[0, 8.944][0, 8.944]	[4, 8.944][0, 7.287]
Temps de calcul moyen	0.3 s	1.8 s

e) Algorithmes d'optimisation par intervalles avec contracteur

Les techniques de contraction que nous avons présentées sont mises en œuvre dans de nombreux algorithmes par intervalles [120, 124, 132], afin de résoudre efficacement le problème d'optimisation globale (6.5). En effet, les contracteurs vont permettre d'isoler dans chaque pavé traité le plus petit sous-pavé qui contient le domaine de faisabilité du problème (6.5) ; la bisection est donc effectuée ensuite sur des pavés plus petits, accélérant ainsi la convergence de l'algorithme. L'intérêt de l'ajout d'un contracteur dans les algorithmes d'optimisation par intervalles est donc le suivant : augmenter le temps de traitement de chaque pavé afin de réduire le temps total de résolution du problème (6.5).

Un modèle d'algorithme par intervalles avec contracteur est illustré sur la Figure 6.14. Il est similaire à l'algorithme de base représenté sur la Figure 6.6 : les étapes principales sont les mêmes à l'exception du test de satisfaction des contraintes qui est remplacé par un contracteur. L'apport du contracteur est double : non seulement il permet de tester la faisabilité du pavé traité (rôle de l'étude

de la satisfaction des contraintes dans l'algorithme de la Figure 6.6), mais en plus il le réduit ou le supprime le cas échéant. Différents algorithmes d'optimisation par intervalles sont comparés dans la partie suivante.

I) Initialisation
1. $L = \{(D, \bar{F})\}, R = \emptyset, M = +\infty, \varepsilon$
II) Recherche du minimum
2. Tant que la liste L n'est pas vide Faire 3. Retirer de L le sous-pavé $[X]$ qui a la plus petite borne inférieure $\underline{F}$ 4. Si $\underline{F} > M$ Alors 5. Supprimer $[X]$ de L 6. Aller à l'étape 2 7. Fin Si
III) Contraction du pavé
8. $[X'] = \text{Contracteur}([X], g_l(X) < 0, l \in \{1, \dots, p\})$ 9. Si $[X'] = \emptyset$ Alors 10. Supprimer $[X]$ de L
IV) Test de la largeur du sous-pavé
11. Sinon 12. Si $w([X]) < \varepsilon$ Alors 13. Ajouter $[X]$ à la liste R
V) Mise à jour de la borne supérieure sur le minimum
14. Calculer la borne supérieure $\bar{F}$ de F sur $[X]$ 15. $M = \min(M, \bar{F})$
VI) Bisection du sous-pavé
16. Sinon 17. Décomposer $[X]$ en deux sous-pavés $[X_1]$ et $[X_2]$ 18. Calculer les bornes inférieures $\underline{F}_1$ et $\underline{F}_2$ de F sur $[X_1]$ et $[X_2]$ 19. Ajouter $([X_1], \underline{F}_1)$ et $([X_2], \underline{F}_2)$ à L 20. Fin Si 21. Fin Si 22. Fin Tant que
VII) Mise à jour de la liste de résultat

23. **Si** la liste R n'est pas vide **Alors**
 24. Supprimer de la liste R les sous-pavés pour lesquels $\underline{F} > M$
 25. **Fin Si**
 26. Retourner la liste R

Figure 6.14 : Algorithme d'optimisation par intervalles avec contracteur

6.3.4 Comparaison des algorithmes d'optimisation par intervalles

La résolution du problème d'optimisation suivant, qui fait intervenir la fonction de Rosenbrock comme fonction de coût, va permettre de comparer les performances des différents algorithmes par intervalles :

$$\begin{aligned}
 &\text{minimiser} && 100(X_2 - X_1^2)^2 + (1 - X_1)^2 \\
 &\text{tel que} && 2 \leq -0.7X_1 + 1.2X_2 - 0.2X_1X_2 \\
 &&& (X_1 - 0.5)^2 + 7(X_2 - 4)^2 \leq 6 \\
 &&& X_1 \in [-3, 4], X_2 \in [2, 6]
 \end{aligned} \tag{6.26}$$

La fonction de coût et le domaine de faisabilité sont représentés sur la Figure 6.15. Le problème d'optimisation (6.26) présente un minimum global (point A) et un minimum local (point B).

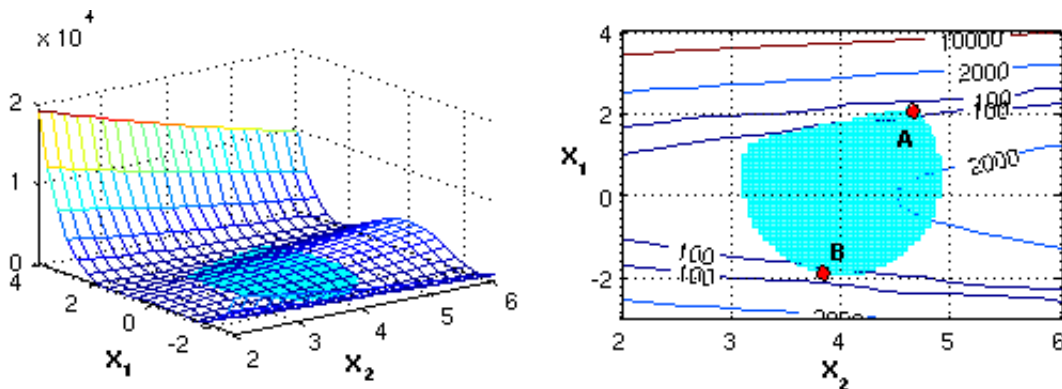


Figure 6.15 : Représentation graphique de la fonction de coût et du domaine de faisabilité (en bleu clair) du problème d'optimisation (6.26). Les points A et B représentent les minima du problème.

L'algorithme par intervalles de base, l'algorithme avec le contracteur par propagation-rétropropagation C_{PR} et l'algorithme avec le contracteur par programmation quadratique C_{PQ} ont été implémentés sous Matlab et appliqués à la résolution du problème d'optimisation (6.26). Pour tous les algorithmes, le critère de terminaison ε qui fixe la taille du plus petit sous-pavé a été fixé à 10^{-3} .

Le minimum min , la valeur de la fonction de coût en ce point f_{min} ainsi que le nombre d'itérations de la boucle principale et le temps de calcul de chaque algorithme sont résumés dans le Tableau 6.2.

Tableau 6.2 : Performances des algorithmes par intervalles pour résoudre le problème (6.26)

Algorithme	min	f_{min}	Nombre d'itérations	Temps de calcul
Algorithme de base	[2.115, 4.486]	1.254	695	51 s
Algorithme avec C_{PR}	[2.114, 4.476]	1.248	389	19 s
Algorithme avec C_{PQ}	[2.114, 4.479]	1.251	349	23 s

Le premier point important à noter est la convergence des trois algorithmes vers le minimum global, c'est-à-dire le point A sur la Figure 6.15. Pour comparaison, la résolution du problème (6.26), en utilisant la fonction « $fmincon$ » de Matlab avec $[-1, 4]$ comme point de départ, donne comme solution le minimum local représenté par le point B , ce qui illustre bien l'intérêt d'outils numériques efficaces pour résoudre les problèmes d'optimisation globale. Concernant la vitesse de convergence, l'algorithme de base est sans surprise le plus lent des trois et celui qui nécessite le plus d'itérations. Sur ce point, l'apport des contracteurs est indéniable : avec le contracteur C_{PR} , le nombre d'itérations est divisé par 1.79 et le temps de calcul par 2.68, tandis qu'avec le contracteur C_{PQ} , le nombre d'itérations est divisé quasiment par 2 et le temps de calcul par 2.22. Il est intéressant de remarquer que l'algorithme avec le contracteur C_{PQ} nécessite moins d'itérations que l'algorithme avec le contracteur C_{PR} , mais que son temps de calcul est supérieur : ceci s'explique par le coût calculatoire du contracteur C_{PQ} qui est plus élevé (cf. §6.3.3d)). Enfin, la Figure 6.16 permet de visualiser l'évolution de la largeur des pavés traités à chaque itération et ainsi de mieux se rendre compte de l'intérêt d'inclure des contracteurs dans les algorithmes d'optimisation par intervalles. L'évolution de la largeur des pavés présente deux phases distinctes : une première phase où les pavés sont réduits jusqu'à ce que le minimum global soit atteint et une deuxième phase où les pavés restants dans la liste L sont éliminés. La Figure 6.16 montre ainsi la décroissance rapide de la largeur des pavés lors de la première phase grâce aux contracteurs, ce qui permet d'isoler rapidement le pavé qui contient le minimum.

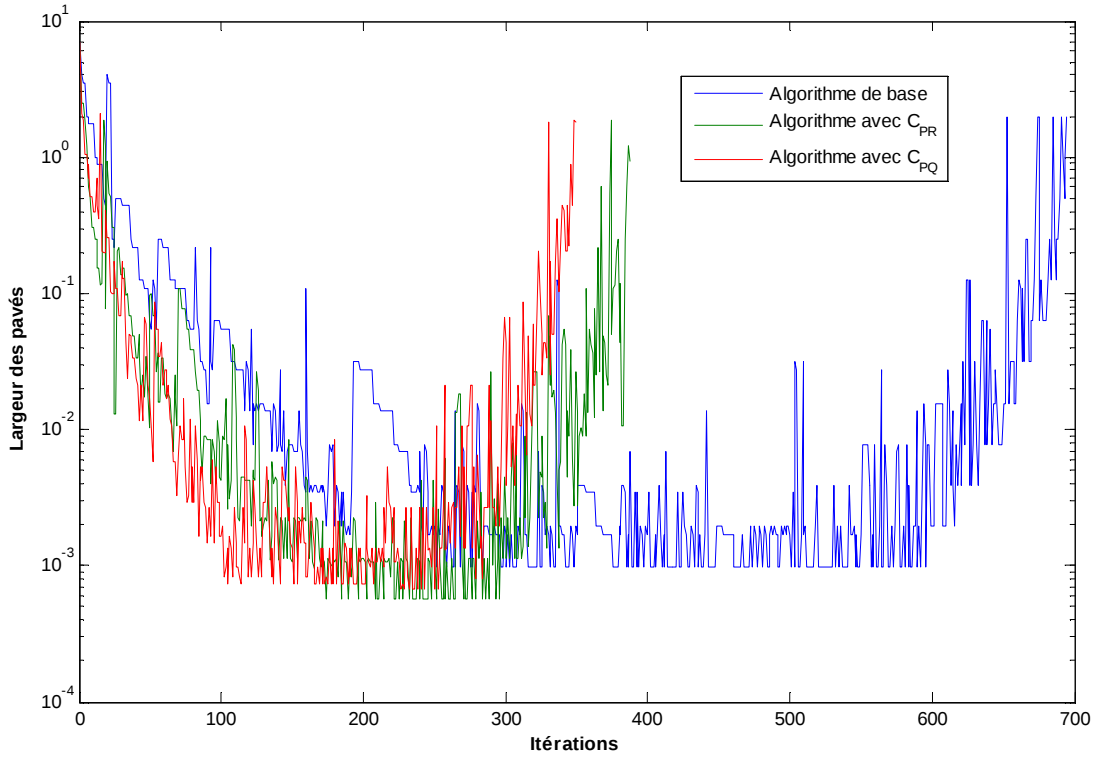


Figure 6.16 : Largeur des pavés traités à chaque itération des algorithmes par intervalles

6.4 Algorithmes par intervalles adaptés à la conception robuste des circuits analogiques

Les algorithmes d'optimisation par intervalles que nous avons présentés dans la partie précédente ont été adaptés afin qu'ils puissent s'appliquer à la conception robuste des circuits analogiques. Le problème d'optimisation (6.4), qui définit la conception robuste, est rappelé ci-dessous :

$$\begin{aligned}
 &\underset{\mathbf{C}}{\text{minimiser}} && \hat{f}_{\max}(\mathbf{C}) \\
 &\text{tel que} && \hat{g}_{l,\max}(\mathbf{C}) < 0, \quad l \in \{1, \dots, p\} \\
 &&& \mathbf{C} \in D
 \end{aligned} \tag{6.27}$$

où $\mathbf{C}$ est le vecteur des paramètres de conception, tandis que $\hat{f}_{\max}(\mathbf{C})$ et $\hat{g}_{l,\max}(\mathbf{C})$ sont les modèles de performances pire-cas. Les modifications qui ont été apportées aux algorithmes par intervalles classiques portent donc sur la mise en place d'une procédure de modélisation de ces performances pire-cas au cours de l'exécution de l'algorithme.

6.4.1 Conception robuste inspirée des algorithmes par intervalles

a) Principe général

La méthode qui a été développée afin d'assurer la conception robuste des circuits analogiques est illustrée sur la Figure 6.17. Cette méthode reprend les principales étapes des algorithmes par intervalles classiques. Deux étapes supplémentaires ont été ajoutées : l'évaluation des valeurs pire-cas de la fonction de coût et des contraintes. Ces étapes mettent en œuvre la méthode de modélisation par morceaux vue au Chapitre 5 et la méthode de modélisation des performances pire-cas expliquée dans la partie §6.2.2, qui fait appel à l'approximation de Cornish-Fisher et aux plans d'expériences.

Évaluation des valeurs pire-cas de la fonction de coût

Cette première étape consiste à construire un modèle par morceaux des valeurs pire-cas de la fonction de coût $\hat{f}_{\max}(C)$, puis à déterminer sur chaque sous-pavé sa valeur maximum $\bar{F}_{\max}$:

- la fonction de coût f est d'abord approximée par un modèle par morceaux $\hat{f}(C, \Delta T)$ sur le domaine d'étude,
- puis un modèle pire-cas $\hat{f}_{\max}(C)$ est construit sur chaque sous-pavé en appliquant la méthode de modélisation des performances pire-cas vue précédemment (cf. 6.2.2),
- enfin la valeur maximum $\bar{F}_{\max}$ de la fonction de coût sur chaque sous-pavé est estimée.

Deux méthodes sont possibles pour évaluer $\bar{F}_{\max}$: l'arithmétique des intervalles ou la résolution d'un problème d'optimisation. En utilisant la fonction d'inclusion naturelle $F_{\max}(C)$ de $\hat{f}_{\max}(C)$, l'arithmétique des intervalles permet d'estimer rapidement la borne supérieure de l'image de $F_{\max}(C)$ sur le sous-pavé. Cependant, le modèle $\hat{f}_{\max}(C)$ est un modèle linéaire avec interactions ou quadratique ; chaque variable apparaît donc plus d'une fois dans le modèle ce qui, d'après le théorème de Moore (voir Annexe C), conduit à une surestimation du maximum. L'autre alternative consiste donc à déterminer précisément le maximum en résolvant le problème d'optimisation suivant :

$$\bar{F}_{\max} = \underset{C \in [C]}{\text{maximiser}} \quad \hat{f}_{\max}(C) \quad (6.28)$$

Evaluation des valeurs pire-cas des contraintes

De la même façon, il est nécessaire d'évaluer les valeurs extrêmes de chaque contrainte sur le sous-pavé $[C]$ qui est traité. Pour cela, le modèle pire-cas des contraintes $\hat{g}_{l,\max}(C)$ doit être construit à partir de la méthode de modélisation des performances pire-cas que nous avons vue au §6.2.2. Si le modèle polynomial des contraintes $\hat{g}_l(C, \mathcal{AT})$ n'existe pas sur le sous-pavé $[C]$, il doit alors être construit avec l'approche séquentielle des plans d'expériences, ce qui nécessite des simulations électriques. Cependant, il est important de noter qu'une fois qu'un modèle polynomial des contraintes valide a été obtenu sur un sous-pavé $[C]$, ce modèle reste valide pour tous les sous-pavés issus de la bisection de $[C]$. Par la suite, il ne sera donc pas nécessaire de le reconstruire à partir de simulations électriques supplémentaires, ce qui permet de réduire le coût en calcul. Enfin, comme pour la fonction de coût, les extrema $\underline{G}_{l,\max}$ et $\overline{G}_{l,\max}$ du modèle pire-cas $\hat{g}_{l,\max}(C)$ sont évalués, en ayant recours à l'arithmétique des intervalles ou par optimisation. Cependant, comme nous l'avons vu, l'arithmétique des intervalles conduit à une surestimation des valeurs extrêmes de $\hat{g}_{l,\max}(C)$ et ainsi à un dimensionnement « sur-robuste ». C'est donc par optimisation que les extrema $\underline{G}_{l,\max}$ et $\overline{G}_{l,\max}$ seront évalués en résolvant les problèmes suivants :

$$\underline{G}_{l,\max} = \underset{C \in [C]}{\text{minimiser}} \quad \hat{g}_{l,\max}(C) \quad (6.29)$$

$$\overline{G}_{l,\max} = \underset{C \in [C]}{\text{maximiser}} \quad \hat{g}_{l,\max}(C) \quad (6.30)$$

I) Initialisation
$L = \emptyset, \varepsilon, \text{ toutes_les_contraintes_satisfaites} = \text{Vrai}$
II) Evaluation des valeurs pire-cas de la fonction de coût
1. Construire un modèle polynomial par morceaux $\hat{f}(C, \mathcal{AT})$ de la fonction de coût f → Liste L de sous-pavés 2. Construire un modèle polynomial $\hat{f}_{\max}(C)$ de la valeur pire-cas de $\hat{f}(C, \mathcal{AT})$ sur chaque sous-pavé de L à partir de l'approximation de Cornish-Fisher et des plans d'expériences 3. Calculer la valeur maximum $\overline{F}_{\max}$ de la performance pire-cas $\hat{f}_{\max}(C)$ sur chaque sous-pavé de L
III) Recherche du minimum

4.	Tant que la liste L n'est pas vide Faire
5.	Retirer de L le sous-pavé $[C]$ qui a la plus petite valeur $\bar{F}_{\max}$
IV) Evaluation des valeurs pire-cas des contraintes	
6.	Pour l allant de 1 à p Faire
7.	Si la contrainte g_l n'a pas de modèle polynomial $\hat{g}_l(C, \mathcal{AT})$ valide sur $[C]$ Alors
8.	Construire un modèle polynomial $\hat{g}_l(C, \mathcal{AT})$ de la contrainte g_l sur $[C]$
9.	Si le modèle $\hat{g}_l(C, \mathcal{AT})$ n'est pas valide Alors
10.	Décomposer $[C]$ en deux sous-pavés $[C_1]$ et $[C_2]$
11.	Calculer les valeurs maximum $\bar{F}_{\max,1}$ et $\bar{F}_{\max,2}$ de $\hat{f}_{\max}(C)$ sur $[C_1]$ et $[C_2]$
12.	Ajouter $([C_1], \bar{F}_{\max,1})$ et $([C_2], \bar{F}_{\max,2})$ à L et retourner à l'étape 4
13.	Fin Si
14.	Fin Si
15.	Construire un modèle polynomial $\hat{g}_{l,\max}(C)$ de la valeur pire-cas de $\hat{g}_l(C, \mathcal{AT})$ sur $[C]$ à partir de l'approximation de Cornish-Fisher et des plans d'expériences
16.	Evaluation des extrema $\underline{G}_{l,\max}$ et $\bar{G}_{l,\max}$ de $\hat{g}_{l,\max}(C)$ sur $[C]$
17.	Fin Pour
Vbis) Contraction du sous-pavé	
18.	$[C'] = \text{Contracteur}([C], \hat{g}_{l,\max}(C) < 0, l \in \{1, \dots, p\})$
19.	Si $[C'] = \emptyset$ Alors
20.	Supprimer $[C]$ de L
21.	Aller à l'étape 4
22.	Fin Si
V) Etude de la satisfaction des contraintes	
23.	Pour l allant de 1 à p Faire
24.	Si $\underline{G}_{l,\max} \geq 0$ Alors
25.	Supprimer $[C]$ de L
26.	Aller à l'étape 4
27.	Sinon Si $\bar{G}_{l,\max} < 0$ Alors
28.	$toutes_les_contraintes_satisfaites = toutes_les_contraintes_satisfaites \& \text{Vrai}$
29.	Sinon
30.	$toutes_les_contraintes_satisfaites = \text{Faux}$
31.	Fin si
32.	Fin Si
33.	Fin Pour
VI) Test de la largeur du sous-pavé	

34.	Si $w([C]) < \varepsilon$ Alors
35.	Si <i>toutes_les_contraintes_satisfaites</i> = Vrai Alors
36.	[C] contient la solution
37.	Fin de l'algorithme
38.	Sinon
39.	Supprimer [C] de L
40.	Fin Si
VII) Bissection du sous-pavé	
41.	Sinon
42.	Décomposer [C] en deux sous-pavés $[C_1]$ et $[C_2]$
43.	Calculer les valeurs maximum $\bar{F}_{\max,1}$ et $\bar{F}_{\max,2}$ de $\hat{f}_{\max}(C)$ sur $[C_1]$ et $[C_2]$
44.	Ajouter $([C_1], \bar{F}_{\max,1})$ et $([C_2], \bar{F}_{\max,2})$ à L
45.	Fin Si
46.	Fin Tant que
47.	Le problème d'optimisation n'a pas de solution

Figure 6.17 : Conception robuste basée sur l'algorithme par intervalles de base

b) Implémentation des méthodes

Deux méthodes de conception robuste ont été implémentées en Java. Les grandes lignes de l'implémentation sont reportées dans l'Annexe D. La première méthode est basée sur l'algorithme par intervalles de base : elle reprend toutes les étapes illustrées sur la Figure 6.17 à l'exception de l'étape Vbis) de contraction. La deuxième utilise le contracteur par programmation quadratique C_{PQ} et implémente l'étape Vbis).

6.4.2 Bilan

Les algorithmes par intervalles classiques ont été modifiés en intégrant la modélisation par morceaux des performances et l'évaluation des valeurs pire-cas basée sur l'approximation de Cornish-Fisher et les plans d'expériences. Ils se révèlent ainsi tout à fait adaptés à la conception robuste. En effet, ils présentent l'avantage de fournir l'optimum global du problème d'optimisation. Ainsi, la meilleure solution est obtenue que les modèles des performances soient convexes ou non sur le domaine d'étude considéré. D'autre part, à chaque itération, un sous-pavé et son modèle de performance sont traités, ce qui se révèle tout à fait adapté à la modélisation par morceaux de la fonction de coût : cette dernière est donc étudiée morceau par morceau. Enfin, il n'est pas nécessaire de modéliser par morceaux les performances qui définissent les contraintes sur l'ensemble du domaine d'étude. En effet, les contraintes ne sont modélisées que sur les sous-pavés susceptibles de contenir

la solution du problème d'optimisation, c'est-à-dire ceux sur lesquels la fonction de coût est minimum. Ainsi, le coût calculatoire de la méthode par intervalles est allégé.

6.5 Application aux circuits analogiques

En appliquant les algorithmes d'optimisation par intervalles à la conception robuste, l'objectif visé est le suivant : déterminer le dimensionnement robuste optimal en prenant en compte la variabilité dès le début, c'est-à-dire sur un domaine de conception relativement large. Les algorithmes par intervalles ont été appliqués à la conception robuste de deux circuits analogiques : un amplificateur opérationnel à transconductance (OTA) et un amplificateur opérationnel de Miller. Les circuits ont été simulés à partir du design kit ST CMOS 65nm développé par STMicroelectronics et du simulateur électrique Eldo. Une station de travail SUN cadencée à 2.8 GHz et équipée de 2 Go de mémoire vive a été utilisée pour les simulations.

6.5.1 Conception robuste d'un amplificateur opérationnel à transconductance

L'amplificateur opérationnel à transconductance est illustré sur la Figure 6.18. Les paramètres de conception ainsi que les variations technologiques considérés sont résumés dans le Tableau 6.3, tandis que les variables d'optimisation sont données dans le Tableau 6.4. Afin d'améliorer la modélisation des performances par des modèles polynomiaux, les largeurs W_i et les longueurs L_i des transistors ne sont pas directement les variables d'optimisation : elles sont remplacées par les variables $A_i = W_i/L_i$ et $B_i = L_i$ qui ont déjà été évoquées dans le Chapitre 5. Deux méthodes de conception robustes basées sur l'algorithme de la Figure 6.17 sont comparées : la première repose sur l'algorithme par intervalles de base, tandis que la deuxième fait appel au contracteur par programmation quadratique C_{PQ} . Les 1^{er} et 99^{ème} centiles sont utilisés pour définir les valeurs pire-cas des performances. Les spécifications de l'OTA sont données dans le Tableau 6.6.

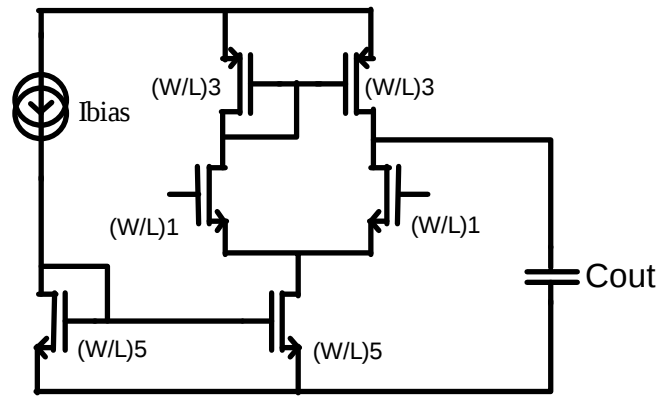


Figure 6.18 : Amplificateur opérationnel à transconductance

Tableau 6.3 : Paramètres de l'amplificateur opérationnel à transconductance

Type de paramètres	Nom	Domaine de variation
Paramètres de conception	W_1	$[1\mu\text{m}, 20\mu\text{m}]$
	L_1	$[1\mu\text{m}, 5\mu\text{m}]$
	W_3	$[1\mu\text{m}, 20\mu\text{m}]$
	L_3	$[1\mu\text{m}, 5\mu\text{m}]$
	W_5	$[1\mu\text{m}, 20\mu\text{m}]$
	L_5	$[1\mu\text{m}, 5\mu\text{m}]$
	I_{bias}	$[15\mu\text{A}, 20\mu\text{A}]$
Variations des paramètres technologiques (variations globales)	Dimensions de la grille	$[-3\sigma, 3\sigma]$
	Dimensions de la zone active	$[-3\sigma, 3\sigma]$
	Epaisseur d'oxyde de grille	$[-3\sigma, 3\sigma]$
	Tension de seuil NMOS	$[-3\sigma, 3\sigma]$
	Tension de seuil PMOS	$[-3\sigma, 3\sigma]$

Tableau 6.4 : Variables d'optimisation

Type de paramètres	Nom	Domaine de variation
Variables d'optimisation	$A_1 = \frac{W_1}{L_1}$	[1, 4]
	$B_1 = L_1$	[1 μ m, 5 μ m]
	$A_3 = \frac{W_3}{L_3}$	[1, 4]
	$B_3 = L_3$	[1 μ m, 5 μ m]
	$A_5 = \frac{W_5}{L_5}$	[1, 4]
	$B_5 = L_5$	[1 μ m, 5 μ m]
	I_{bias}	[15 μ A, 20 μ A]

Coût calculatoire

Le coût calculatoire de chaque méthode est indiqué dans le Tableau 6.5. La première étape des deux méthodes consiste à construire des modèles polynomiaux de la fonction de coût et des contraintes à partir des simulations Eldo ; cette étape dure un peu moins de 10 min. Une fois qu'un modèle valide existe sur chaque sous-pavé traité, la recherche de l'optimum commence. L'algorithme par intervalles avec le contracteur quadratique se révèle alors bien plus efficace que l'algorithme de base : le temps de calcul est quasiment divisé par quatre.

Tableau 6.5 : Comparaison du coût calculatoire des deux méthodes

Méthode	Nombre de simulations Eldo	Nombre d'itérations	Temps de calcul
Algorithme par intervalles de base	1035	32222	115 min
Algorithme par intervalles avec contracteur C_{PQ}	1035	1806	31 min

Robustesse

Les deux méthodes convergent vers le même optimum. Afin de vérifier la robustesse du circuit, une analyse de Monte Carlo (1000 simulations électriques) est effectuée à cet optimum. Les distributions des performances sont illustrées sur la Figure 6.19(a)-(f) et les valeurs pire-cas estimées

avec l'analyse de Monte Carlo sont résumées dans le Tableau 6.6. Les résultats obtenus confirment la robustesse du dimensionnement vis-à-vis des dispersions paramétriques.

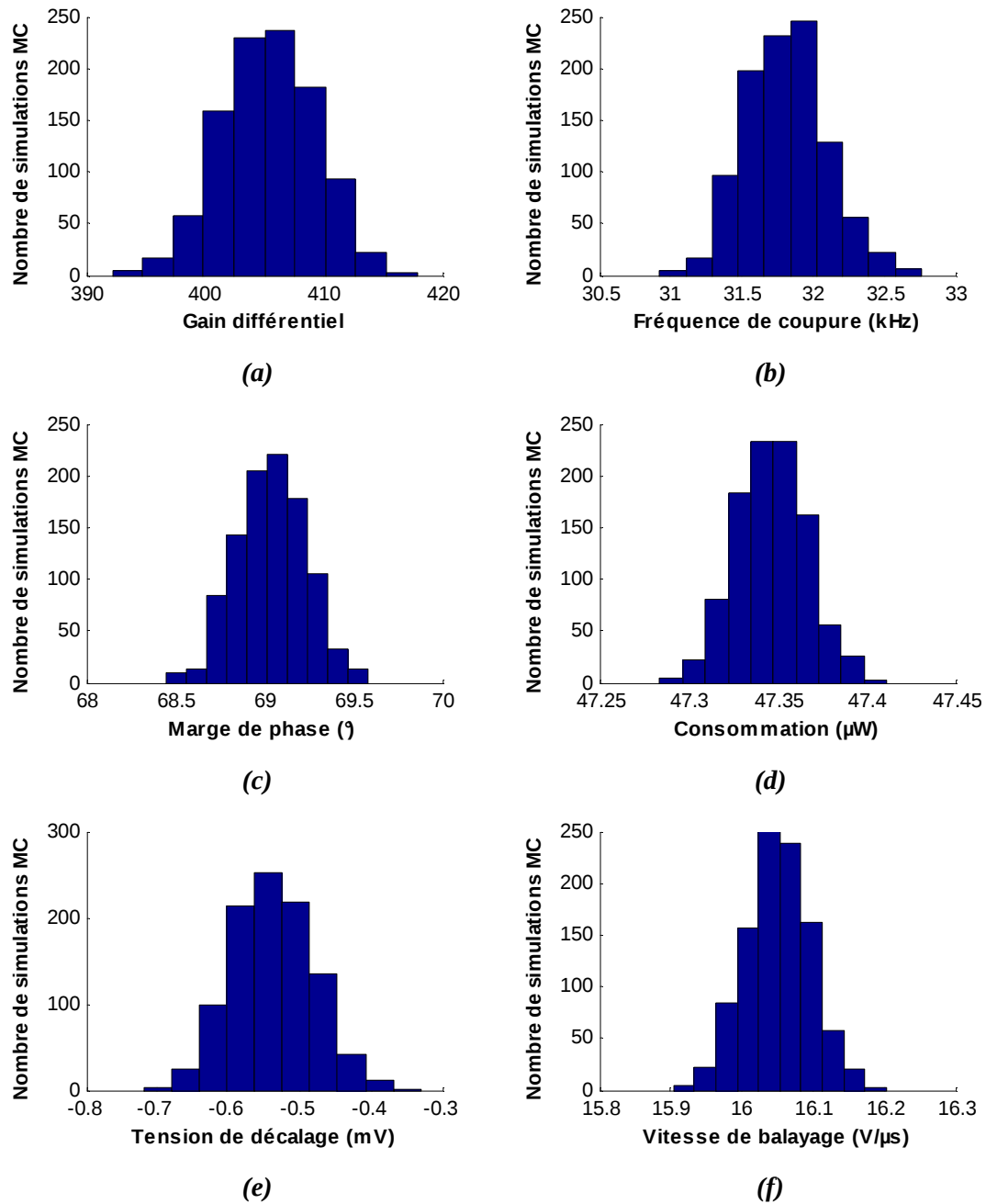


Figure 6.19 : Distributions des performances de l'OTA au point de conception robuste

Tableau 6.6 : Résultats de la conception robuste de l'OTA

Performances	Gain	Fréquence de coupure	Marge de phase	Consommation	Tension de décalage	Vitesse de balayage
Spécifications	Maximiser	> 10kHz	> 67°	< 50μW	< 1mV	> 15V/μs
Performance pire-cas	392	31kHz	68.4°	47.4μW	0.72mV	15.9V/μS

a) Réduction du domaine de conception : méthode de dimensionnement gm/I_D

L'exemple précédent a permis de mettre en avant le rôle important des contracteurs pour diminuer le coût calculatoire. Ce coût pourrait encore être considérablement diminué s'il était possible de réduire rapidement la taille du domaine de conception initial. Cela peut être réalisé en intégrant des contraintes implicites en plus des spécifications du circuit.

b) Réduction du domaine de conception

Une première approche consiste à appliquer des règles de pré-dimensionnement portant sur la polarisation des transistors ainsi que leur zone de fonctionnement et basées uniquement sur des simulations statiques DC [133]. Afin de réduire le coût calculatoire, une alternative peut consister à remplacer les simulations électriques par des équations analytiques. Le principe est alors d'intégrer dans le problème d'optimisation robuste des contraintes supplémentaires inspirées de la conception manuelle afin d'identifier grossièrement mais facilement le domaine de faisabilité. Les équations analytiques peuvent être des modèles simplifiés des transistors (Spice Level 1 par exemple [134]) qui permettent d'approximer les performances et ainsi obtenir une estimation grossière du domaine de faisabilité. Cependant, ces modèles simplifiés dépendent fortement de la zone de fonctionnement des transistors. De plus, ils ne permettent pas de tenir compte des dispersions paramétriques. Nous avons donc mis en place dans notre méthode de conception robuste une approche qui est fondée sur le rapport gm/I_D où gm est le gain de transconductance d'un transistor et I_D son courant de drain.

c) Méthode de dimensionnement gm/I_D

La méthode de dimensionnement gm/I_D a été proposée par [135] pour concevoir des circuits analogiques de faible consommation et intégrée dans des outils de synthèse automatique d'amplificateurs [136]. Cette méthode repose sur la relation particulière qui unit le rapport gm/I_D au courant normalisé $I_D/(W/L)$ où W et L représentent respectivement la largeur et la longueur d'un transistor. En effet, le principal intérêt de cette relation est qu'elle caractérise de façon unique tous les transistors du même type (NMOS ou PMOS) appartenant à la même technologie CMOS quelles que soient leurs dimensions. De plus, cette relation permet de représenter toutes les zones de fonctionnement des transistors (inversion faible, modérée ou forte). Enfin, la plupart des performances des circuits analogiques peuvent s'exprimer aisément en fonction du rapport gm/I_D .

La courbe qui représente la relation entre le rapport gm/I_D et le courant normalisé $I_D/(W/L)$ peut être obtenue par simulation électrique à partir d'un modèle de transistor continu entre les zones de fonctionnement et qui caractérise la technologie CMOS utilisée. La courbe

$gm/I_D = f(I_D/(W/L))$ caractéristique d'une technologie CMOS 65nm développée par STMicroelectronics est illustrée sur la Figure 6.20. Dans le cas des transistors NMOS, $I_D = I_{DS}$ et dans le cas des transistors PMOS, $I_D = |I_{DS}|$. La différence entre les courbes des transistors NMOS et PMOS s'explique par leur mobilité respective qui est différente.

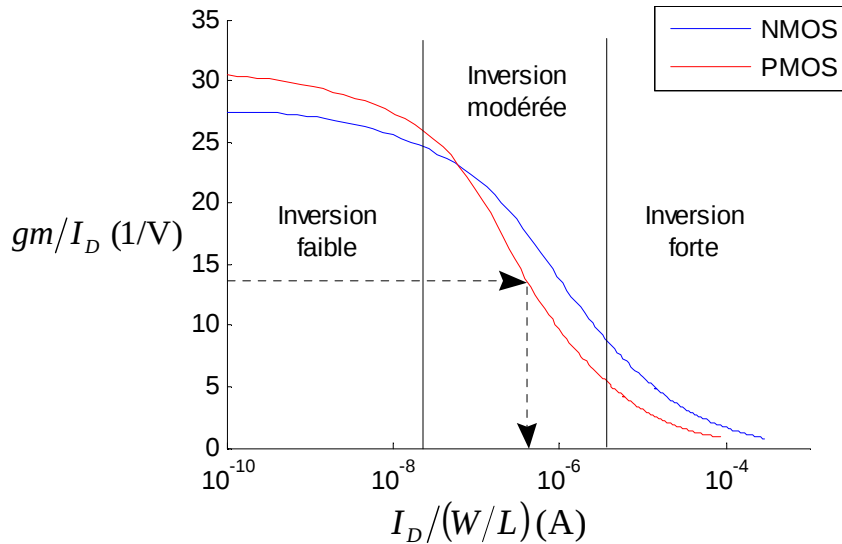


Figure 6.20 : Caractéristique $gm/I_D = f(I_D/(W/L))$ d'une technologie CMOS 65nm

Grâce à cette relation qui caractérise de façon unique une technologie CMOS, les circuits analogiques peuvent être dimensionnés de façon rapide et précise :

- en exprimant d'abord les performances en fonction du rapport gm/I_D des transistors,
- puis en ajustant la valeur du rapport gm/I_D des transistors pour que les spécifications soient satisfaites,
- enfin en utilisant la courbe $gm/I_D = f(I_D/(W/L))$ pour déterminer les dimensions des transistors à partir de leur rapport gm/I_D et de leur courant de drain I_D .

d) Amplificateur opérationnel de Miller

L'exemple suivant va permettre d'illustrer cette méthode en l'appliquant au dimensionnement d'un amplificateur opérationnel de Miller (cf. Figure 6.21). Le dimensionnement est réalisé à partir des spécifications sur le gain différentiel, le produit gain-bande passante, la marge de phase et la vitesse de balayage respectivement notées A_{spec} , GBW_{spec} , MP_{spec} et SR_{spec} . Les étapes de dimensionnement sont décrites ci-dessous. Elles visent à déterminer le rapport gm/I_D et le courant I_D pour chaque transistor.

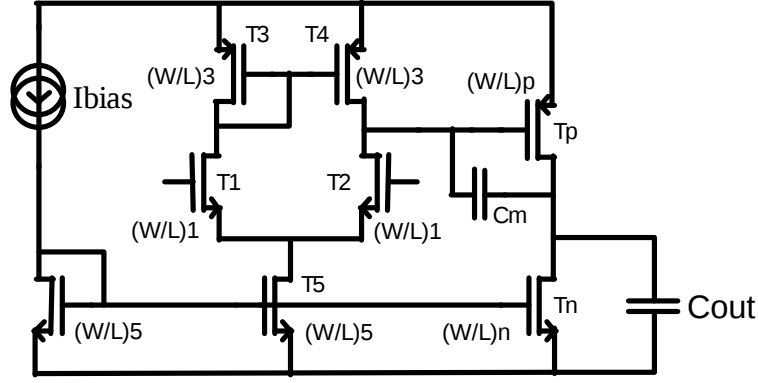


Figure 6.21 : Amplificateur opérationnel de Miller

1. La valeur de la capacité de Miller C_m est d'abord choisie de façon à satisfaire une marge de phase supérieure à 67° [134] :

$$C_m \geq \frac{3}{10} C_{out} \quad (6.31)$$

2. Le courant de polarisation I_{bias} est ensuite fixé afin que la spécification sur la vitesse de balayage soit respectée :

$$I_{bias} \geq SR_{spec} \cdot C_m \quad (6.32)$$

3. Les transistors T3 et T4, qui forment un miroir de courant, doivent opérer en régime de forte inversion (cf. Figure 6.20) pour garantir un bon appariement :

$$\frac{gm_3}{I_{D3}} < 6 \quad \text{avec} \quad I_{D3} = \frac{I_{bias}}{2} \quad (6.33)$$

4. Afin que les transistors T3, T4 fonctionnent bien en miroir de courant, le transistor T4 doit être en régime de saturation, ce qui peut être rendu possible en imposant $V_{SG4} = V_{SD4} = V_{SGp}$. Le transistor Tp vérifie alors :

$$\frac{gm_p}{I_{Dp}} = \frac{gm_3}{I_{D3}} \quad (6.34)$$

5. De plus, les conditions sur les pôles pour avoir une marge de phase supérieure à 67° conduisent à [134] :

$$gm_p \geq 10 \cdot GBW_{spec} \cdot 2\pi \cdot C_m \Rightarrow I_{Dp} \geq \frac{10 \cdot GBW_{spec} \cdot 2\pi \cdot C_m}{gm_3 / I_{D3}} \quad (6.35)$$

6. Pour que la spécification sur la bande passante soit respectée, il est nécessaire que [134] :

$$gm_1 \geq GBW_{spec} \cdot 2\pi \cdot C_m \quad \text{avec} \quad I_{D1} = \frac{I_{bias}}{2} \quad (6.36)$$

7. Enfin, les conditions pour obtenir un bon appariement entre les transistors du miroir de courant T5 et Tn imposent (régime de forte inversion) :

$$\frac{gm_5}{I_{D5}} = \frac{gm_n}{I_{Dn}} < 8 \quad \text{avec} \quad I_{D5} = I_{bias} \quad \text{et} \quad I_{Dn} = I_{Dp} \quad (6.37)$$

A présent que les rapports gm/I_D sont fixés pour chaque transistor, la courbe $gm/I_D = f(I_D/(W/L))$ permet de déterminer les valeurs $I_D/(W/L)$ correspondantes. Les valeurs des courants I_D étant fixées également, on en déduit alors le rapport des dimensions W/L pour chaque transistor. Les règles de dimensionnement (6.31)-(6.37) garantissent que le circuit opère dans une zone de fonctionnement stable lui permettant d'atteindre les spécifications. Elles peuvent donc être rajoutées au problème de conception robuste sous la forme de contraintes implicites afin de réduire le domaine de conception.

6.5.2 Conception robuste d'un amplificateur opérationnel de Miller

La méthode de dimensionnement gm/I_D a été intégrée dans l'algorithme de conception robuste par intervalles afin d'accélérer sa convergence. Le principe, illustré sur la Figure 6.22, consiste à utiliser :

- les rapports gm/I_D comme variables d'optimisation,
- les règles de dimensionnement (6.31)-(6.37) pour réduire le domaine de conception,
- enfin la courbe $gm/I_D = f(I_D/(W/L))$ pour faire le lien avec les dimensions des transistors et ainsi réaliser des simulations électriques permettant de prendre en compte les dispersions paramétriques pour construire les modèles des performances pire-cas.

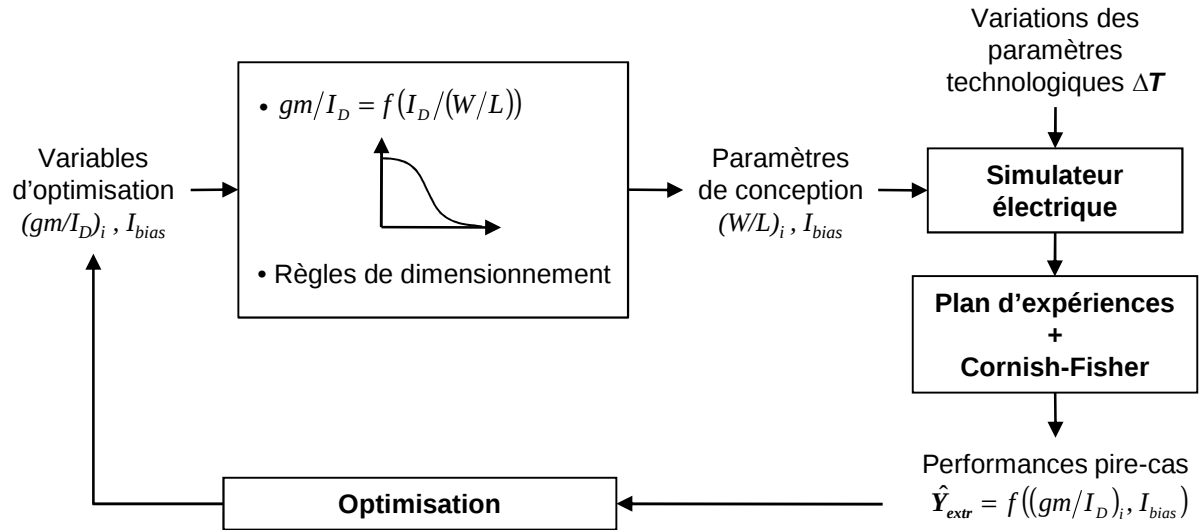


Figure 6.22 : Intégration de la méthode de dimensionnement gm/I_D au sein de l'algorithme de conception robuste par intervalles

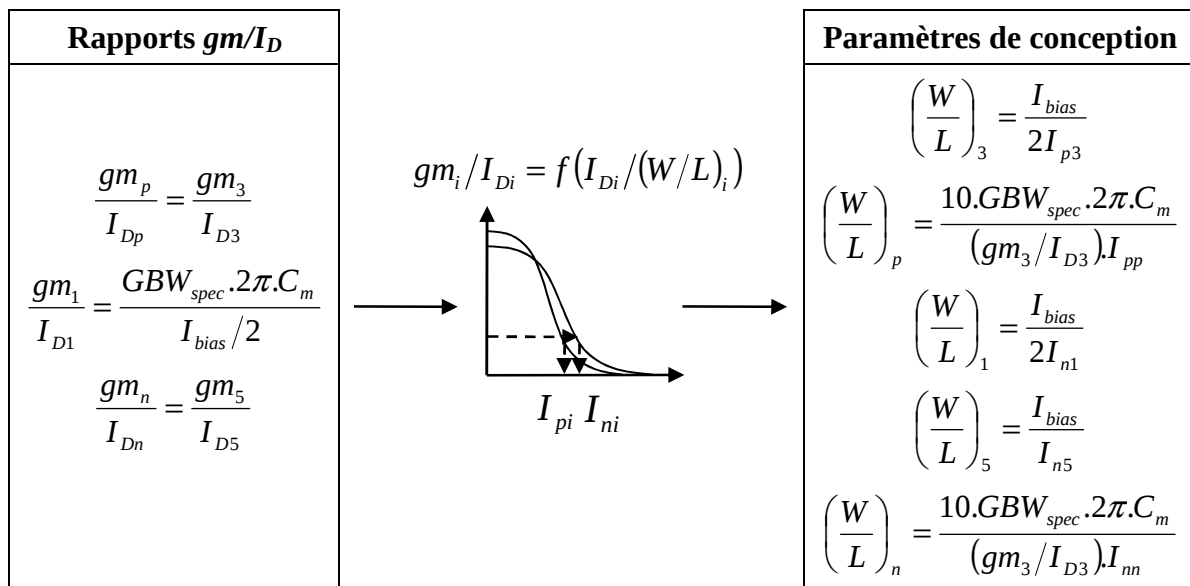
Les variations paramétriques prises en compte sont présentées dans le Tableau 6.7, tandis que les variables d'optimisation sont données dans le Tableau 6.8. Les domaines de variation de la capacité de Miller C_m et du courant de polarisation I_{bias} sont issus des règles de dimensionnement (6.31) et (6.32) sur la stabilité et la vitesse de balayage, tandis que ceux des rapports gm_3/I_{D3} et gm_5/I_{D5} permettent de garantir un fonctionnement des transistors en forte inversion. A partir de ces variables d'optimisation et des règles de dimensionnement (6.31)-(6.37), les rapports gm/I_D de tous les transistors peuvent être calculés, puis en utilisant la courbe $gm/I_D = f(I_D/(W/L))$, les valeurs des paramètres de conception peuvent être extraites comme illustré sur la Figure 6.23. Les valeurs I_{ni} et I_{pi} , qui correspondent aux courants normalisés $I_{Di}/(W/L)_i$ lus sur la courbe $gm/I_D = f(I_D/(W/L))$ pour chaque transistor i , permettent de calculer les dimensions $(W/L)_i = I_{Di}/I_{ni}$ dans le cas d'un transistor NMOS et $(W/L)_i = I_{Di}/I_{pi}$ dans le cas d'un transistor PMOS.

Tableau 6.7 : Variations des paramètres technologiques de l'amplificateur opérationnel de Miller

Type de paramètres	Nom	Domaine de variation
Variations des paramètres technologiques (variations globales)	Dimensions de la grille	$[-3\sigma, 3\sigma]$
	Dimensions de la zone active	$[-3\sigma, 3\sigma]$
	Epaisseur d'oxyde de grille	$[-3\sigma, 3\sigma]$
	Tension de seuil NMOS	$[-3\sigma, 3\sigma]$
	Tension de seuil PMOS	$[-3\sigma, 3\sigma]$
	Dimensions de capacité Poly-Poly	$[-3\sigma, 3\sigma]$

Tableau 6.8 : Variables d'optimisation

Type de paramètres	Nom	Domaine de variation
Variables d'optimisation	C_m	$\left[\frac{3}{10} C_{out}, C_{max} \right]$
	I_{bias}	$[SR_{spec} \cdot C_m, I_{max}]$
	$\frac{gm_3}{I_{D3}}$	$]0, 6]$
	$\frac{gm_5}{I_{D5}}$	$]0, 8]$

**Figure 6.23 : Extraction des valeurs des paramètres de conception à partir des rapports gm/I_D**

L'algorithme par intervalles avec le contracteur quadratique est utilisé pour la conception robuste de l'amplificateur de Miller. Le coût calculatoire est donné dans le Tableau 6.9. Grâce aux règles de dimensionnement gm/I_D , le temps de calcul pour dimensionner l'amplificateur de Miller reste raisonnable. Concernant la robustesse, une analyse de Monte Carlo (1000 simulations électriques) est effectuée afin d'estimer les valeurs pire-cas (cf. Tableau 6.10). Les résultats obtenus indiquent que les valeurs pire-cas respectent les spécifications ; la robustesse du circuit est donc garantie sans pour autant nécessiter un surdimensionnement.

Tableau 6.9 : Coût calculatoire de la conception robuste de l'amplificateur de Miller

Méthode	Nombre de simulations Eldo	Nombre d'itérations	Temps de calcul
Algorithme par intervalles avec le contracteur C_{PQ}	773	999	22 min

Tableau 6.10 : Résultats de la conception robuste de l'amplificateur de Miller

Performances	Consommation	Gain	Produit gain-bande passante	Marge de phase	Tension de décalage	Vitesse de balayage
Spécifications	Minimiser	$> 10^4$	$> 10\text{MHz}$	$> 67^\circ$	$< 1\text{mV}$	$> 10\text{V}/\mu\text{s}$
Performance pire-cas	$457\mu\text{W}$	1.4×10^4	12.2MHz	68.3°	0.05mV	$11.8\text{V}/\mu\text{S}$

6.6 Conclusion

Dans ce chapitre, nous avons présenté une nouvelle méthode de conception robuste basée sur des algorithmes d'optimisation par intervalles. Elle a pour objectif de déterminer les valeurs des paramètres de conception pour lesquelles le circuit sera robuste aux dispersions paramétriques et respectera les spécifications. Cette approche intègre l'approximation de Cornish-Fisher pour estimer les performances pire-cas qui sont comparées aux spécifications pour tester la robustesse du circuit de façon précise et efficace. Grâce à l'approximation de Cornish-Fisher, les performances pire-cas sont définies pour un rendement paramétrique donné, ce qui permet d'éviter un surdimensionnement (comme c'est le cas avec les analyses pire-cas). Par ailleurs, les algorithmes par intervalles garantissent une optimisation globale et se révèlent tout à fait adaptés pour traiter les modèles par morceaux des performances. Le coût de calcul de la méthode résulte du coût de calcul de la modélisation des performances et du coût de calcul de l'algorithme par intervalles. Comme nous l'avons vu dans le

Chapitre 4, le coût de construction des modèles polynomiaux à partir des plans d'expériences devient prohibitif au-delà de vingt facteurs, ce qui constitue la principale limitation en termes de nombre de paramètres pouvant être pris en compte dans la méthode. Concernant le coût de calcul de l'algorithme par intervalles, il peut être réduit en mettant en œuvre, d'une part des techniques mathématiques de réduction d'intervalles afin d'accélérer la convergence de l'algorithme, et d'autre part des règles de dimensionnement issues de la conception manuelle permettant de diminuer la taille du domaine de conception. Les règles de dimensionnement gm/I_D s'avèrent particulièrement intéressantes dans la mesure où elles permettent de combiner le savoir-faire du concepteur de circuits analogiques avec la précision du simulateur électrique qui rend possible la prise en compte des variations paramétriques. De plus, le découpage du domaine initial offre la possibilité de traiter en priorité les sous-pavés les plus probables de contenir l'optimum et donc de n'approximer les contraintes que sur ces sous-pavés, ce qui économise ainsi de nombreuses simulations électriques. Grâce à ces méthodes de réduction du coût calculatoire, l'algorithme par intervalles peut être appliqué sur des domaines relativement larges, permettant ainsi de tenir compte des dispersions paramétriques dès le début du flot de conception. Les résultats obtenus sur des circuits analogiques montrent l'efficacité de la méthode pour dimensionner de façon automatisée des circuits robustes sans surdimensionnement.

Chapitre 7 : Conclusion et perspectives

Dans ce travail de thèse, nous avons étudié les phénomènes de dispersions paramétriques dans les technologies CMOS, ainsi que les méthodes permettant de tenir compte de l'impact de ces dispersions lors de la conception des circuits analogiques. Deux familles de méthodes peuvent être distinguées : les méthodes d'analyse de la variabilité et les méthodes de conception robuste. Les premières visent à caractériser les dispersions des performances à partir de mesures statistiques comme le rendement de fabrication ou les valeurs pire-cas, tandis que les deuxièmes vont plus loin en intégrant ces méthodes d'analyse de la variabilité dans un processus d'optimisation afin de rendre les circuits robustes aux dispersions paramétriques.

7.1 Analyse de la variabilité par les plans d'expériences et le développement limité de Cornish-Fisher

Les méthodes d'analyse de la variabilité classiques se révèlent soit imprécises (analyse pire-cas), soit lourdes en termes de coût calculatoire (analyse de Monte Carlo). Dans le Chapitre 4, une nouvelle méthode d'analyse a été proposée : elle vise à estimer les valeurs pire-cas des performances. Les performances sont d'abord approximées par des modèles linéaires avec interactions ou des modèles quadratiques en faisant appel aux plans d'expériences. Ces modèles polynomiaux présentent plusieurs avantages : ils sont facilement interprétables et peuvent être construits à moindre coût grâce aux plans d'expériences tant que le nombre de paramètres en jeu reste faible. Enfin, les valeurs pire-cas sont estimées avec le développement limité de Cornish-Fisher et les moments des performances calculés à partir des modèles polynomiaux. Les résultats obtenus sur des circuits analogiques de base montrent que cette approche présente un bon compromis précision/coût calculatoire et peut donc parfaitement s'intégrer dans les méthodes de conception robuste.

7.2 Modélisation par morceaux des performances

Avec la miniaturisation, le nombre et l'amplitude des dispersions paramétriques augmentent. Les modèles quadratiques peuvent alors se révéler insuffisants pour approximer les performances des

circuits. Dans le Chapitre 5, une méthode de modélisation par morceaux a été présentée. Son objectif est d'approximer les non-linéarités des performances sur un domaine d'étude relativement large en le découpant de façon judicieuse en sous-domaines sur lesquels les performances tendent à être moins non-linéaires. L'approximation obtenue permet de définir le comportement global du circuit et ainsi identifier les domaines d'intérêt pour la conception.

7.3 Optimisation robuste par intervalles

Les méthodes de modélisation par les plans d'expériences et d'estimation des valeurs pire-cas avec l'approximation de Cornish-Fisher ont été intégrées dans un algorithme d'optimisation globale par intervalles afin d'automatiser la conception de circuits analogiques robustes. Les algorithmes par intervalles, en optimisant les performances pire-cas, garantissent de trouver le meilleur dimensionnement robuste sur le domaine d'étude considéré. De plus, en raisonnant sur des intervalles plutôt que sur des valeurs ponctuelles, ils se révèlent parfaitement adaptés pour traiter les modèles par morceaux des performances. Cependant, la mise en place de cette approche de conception robuste dès le début du flot de conception augmente considérablement le temps de calcul. Des techniques basées sur des contracteurs ou des règles de dimensionnement analogique sont donc appliquées pour identifier rapidement le domaine de faisabilité des circuits.

7.4 Perspectives

Dans ce travail de thèse, seules les variations globales ont été prises en compte dans l'optimisation robuste par intervalles ; les variations environnementales et locales doivent également être considérées. Concernant les variations des conditions environnementales comme la température ou la tension d'alimentation, elles peuvent être prises en compte en les modélisant sous la forme d'intervalles de tolérance sur lesquels les performances pire-cas sont calculées de la même façon que pour les variations paramétriques. Quant aux variations locales, la principale difficulté réside dans leur nombre qui augmente proportionnellement avec le nombre de composants. Des méthodes d'identification des variations locales les plus significatives sont donc nécessaires. En ce qui concerne le développement limité de Cornish-Fisher, qui s'est révélé efficace pour estimer les performances pire-cas, deux points nécessitent d'être approfondis : l'impact de l'ordre des cumulants sur la précision du développement et la qualité de l'estimation des centiles extrêmes. La méthode de conception robuste que nous avons développée est destinée aux circuits analogiques élémentaires. Par la suite, il semble donc important de l'étendre à des systèmes plus complexes. L'intérêt des techniques de propagation d'intervalles, pour répartir les spécifications entre des blocs de niveaux hiérarchiques différents, a déjà été démontré dans [121] ; cette approche pourrait être

complétée par les méthodes d'analyse de la variabilité que nous avons proposées dans cette thèse afin de réaliser une conception hiérarchique robuste. En ce qui concerne les plans d'expériences, ils ont été appliqués pour construire des méta-modèles qui sont utilisés lors de la conception des circuits. Ces mêmes méta-modèles, établis entre des performances mesurables et les variations paramétriques, pourraient permettre également d'analyser la variabilité lors du fonctionnement des circuits. On pourrait alors envisager, par exemple, un système de compensation de la variabilité à base de capteurs et de correcteurs, ces derniers étant étalonnés à partir des modèles établis avec les méthodes présentées dans cette thèse.

Les innovations des outils de CAO visent généralement à atteindre le meilleur compromis précision/temps de calcul. Une première approche pour réduire le coût calculatoire consiste à réduire le nombre de variables en jeu. Le développement des techniques de réduction de dimension et leur incorporation dans les outils de CAO est donc plus que jamais nécessaire. Il est également important d'ajuster la précision et le coût de construction des méta-modèles en fonction du problème traité. Dans cette optique, les méthodes dites « Metamodel-based Design Optimization » [137] qui intègrent le processus de modélisation au cœur d'un algorithme d'optimisation sont particulièrement intéressantes : elles ont pour but de concentrer les points des plans d'expériences autour de l'optimum permettant ainsi d'accroître la précision du méta-modèle seulement là où c'est nécessaire. Enfin, la réduction du coût calculatoire n'est pas un problème auquel seules les mathématiques appliquées peuvent apporter une solution : comme nous l'avons démontré, l'incorporation dans les outils de CAO du savoir-faire du concepteur de circuits analogiques permet également de raccourcir efficacement les temps de calcul. Il semblerait donc intéressant d'approfondir ces approches qui combinent à la fois des techniques mathématiques innovantes et des méthodes de dimensionnement inspirées de la conception manuelle.

Bibliographie

- [1] G. Moore, “Cramming more components onto integrated circuits,” *Electronics*, vol. 38, pp. 114–117, April 1965.
- [2] G. Moore, “Progress in digital integrated electronics,” in *Proceedings of the International Electron Devices Meeting*, vol. 21, pp. 11–13, 1975.
- [3] The International Technology Roadmap for Semiconductors (ITRS). <http://www.itrs.net>.
- [4] R. Dennard, F. Gaensslen, V. Rideout, E. Bassous, and A. LeBlanc, “Design of ion-implanted MOSFET’s with very small physical dimensions,” *IEEE Journal of Solid-State Circuits*, vol. 9, no. 5, pp. 256–268, 1974.
- [5] T. Terasawa, “Subwavelength lithography (PSM, OPC),” in *Proceedings of the Asia and South Pacific Design Automation Conference (ASP-DAC)*, pp. 295–300, 2000.
- [6] B. P. Wong, A. Mittal, Y. Cao, and G. Starr, *Nano-CMOS Circuit and Physical Design*. Wiley, 2004.
- [7] K. Roy, S. Mukhopadhyay, and H. Mahmoodi-Meimand, “Leakage current mechanisms and leakage reduction techniques in deep-submicrometer CMOS circuits,” *Proceedings of the IEEE*, vol. 91, no. 2, pp. 305–327, 2003.
- [8] T. Burd and R. Brodersen, “Energy efficient CMOS microprocessor design,” in *Proceedings of the Twenty-Eighth Hawaii International Conference on System Sciences*, vol. 1, pp. 288–297, 1995.
- [9] S. Miermont, P. Vivet, and M. Renaudin, “A power supply selector for energy- and area-efficient local dynamic voltage scaling,” in *Proceedings of the 17th international workshop on Integrated Circuit and System Design. Power and Timing Modeling, Optimization and Simulation (PATMOS)*, vol. 4644, pp. 556–565, 2007.
- [10] K. Bernstein, P. Andry, J. Cann, P. Emma, D. Greenberg, W. Haensch, M. Ignatowski, S. Koester, J. Magerlein, R. Puri, and A. Young, “Interconnects in the third dimension: design challenges for 3D ICs,” in *Proceedings of the 44th annual Design Automation Conference (DAC)*, pp. 562–567, 2007.

- [11] I. O'Connor, F. Tissafi-Drissi, F. Gaffiot, J. Dambre, M. De Wilde, J. Van Campenhout, D. Van Thourhout, and D. Stroobandt, "Systematic simulation-based predictive synthesis of integrated optical interconnect," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 15, no. 8, pp. 927–940, 2007.
- [12] I. O'Connor, J. Liu, F. Gaffiot, F. Pregaldiny, C. Lallement, C. Maneux, J. Goguët, S. Fregonese, T. Zimmer, L. Anghel, T.-T. Dang, and R. Leveugle, "CNTFET modeling and reconfigurable logic-circuit design," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 54, no. 11, pp. 2365–2379, 2007.
- [13] A. Jalabert, F. Clermidy, and A. Amara, "A non-volatile multi-level memory cell using molecular-gated nanowire transistors," in *Proceeding of the 13th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, pp. 1034–1037, 2006.
- [14] E. Ollier, P. Andreucci, L. Duraffourg, E. Colinet, C. Durand, F. Casset, T. Ernst, S. Hentz, S. Labarthe, C. Marcoux, V. Nguyen, D. Renaud, E. Mile, P. Renaux, D. Mercier, P. Robert, P. Ancey, and L. Bouchailot, "NEMS based on top-down technologies: from stand-alone NEMS to VLSI NEMS & NEMS-CMOS integration," in *Proceedings of the IEEE International Conference on Electron Devices and Solid-State Circuits (EDSSC)*, pp. 1–6, 2008.
- [15] D. Lattard, E. Beigne, F. Clermidy, Y. Durand, R. Lemaire, P. Vivet, and F. Berens, "A reconfigurable baseband platform based on an asynchronous network-on-chip," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 1, pp. 223–235, 2008.
- [16] R. Raina, "What is DFM & DFY and Why Should I Care ?," in *Proceedings of the IEEE International Test Conference (ITC)*, pp. 1–9, 2006.
- [17] C. Saunier, "L'industrie de la microélectronique : reprendre l'offensive," tech. rep., Office parlementaire d'évaluation des choix scientifiques et technologiques, 2008.
- [18] K. Bernstein, D. Frank, A. Gattiker, W. Haensch, B. Ji, S. Nassif, E. Nowak, D. Pearson, and N. Rohrer, "High-performance CMOS variability in the 65-nm regime and beyond," *IBM Journal of Research and Development*, vol. 50, no. 4/5, pp. 433–449, 2006.
- [19] A. Srivastava, D. Sylvester, and D. Blaauw, *Statistical analysis and optimization for VLSI*. Springer, 2005.
- [20] S. Nassif, "Design for variability in DSM technologies," in *Proceedings of the IEEE International Symposium on Quality Electronic Design (ISQED)*, pp. 451–454, 2000.

- [21] H. Elzinga, “On the impact of spatial parametric variations on MOS transistor mismatch,” in *Proceedings of the IEEE International Conference on Microelectronic Test Structures (ICMTS)*, pp. 173–177, 1996.
- [22] A. Asenov, A. Brown, J. Davies, S. Kaya, and G. Slavcheva, “Simulation of intrinsic parameter fluctuations in decananometer and nanometer-scale MOSFETs,” *IEEE Transactions on Electron Devices*, vol. 50, no. 9, pp. 1837–1852, 2003.
- [23] J.-B. Shyu, G. Temes, and K. Yao, “Random errors in MOS capacitors,” *IEEE Journal of Solid-State Circuits*, vol. 17, no. 6, pp. 1070–1076, 1982.
- [24] S. Nassif, “Process variability at the 65nm node and beyond,” in *Proceedings of the IEEE Custom Integrated Circuits Conference (CICC)*, pp. 1–8, 2008.
- [25] Electronics Research Laboratory of the University of California at Berkeley, “BSIM.” <http://www-device.eecs.berkeley.edu/~bsim3/>.
- [26] Arizona State University and NXP Semiconductors Research, “PSP.” <http://pspmodel.asu.edu>.
- [27] Ecole Polytechnique Fédérale de Lausanne (EPFL), “EKV.” <http://legwww.epfl.ch/ekv/>.
- [28] E. Felt, S. Zanella, C. Guardiani, and A. Sangiovanni-Vincentelli, “Hierarchical statistical characterization of mixed-signal circuits using behavioral modeling,” in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 374–380, 1996.
- [29] C. Guardiani, S. Saxena, P. McNamara, P. Schumaker, and D. Coder, “An asymptotically constant, linearly bounded methodology for the statistical simulation of analog circuits including component mismatch effects,” in *Proceedings of the 37th Design Automation Conference (DAC)*, pp. 15–18, 2000.
- [30] Y. Dupret, Modélisation des dispersions des performances des cellules analogiques intégrées en fonction des dispersions du processus de fabrication. PhD thesis, Université Paris-Sud - Supélec, 2005.
- [31] A. Mitev, M. Marefat, D. Ma, and J. Wang, “Parameter reduction for variability analysis by slice inverse regression (SIR) method,” in *Proceedings of the Asia and South Pacific Design Automation Conference (ASP-DAC)*, pp. 468–473, 2007.
- [32] M. Pelgrom, A. Duinmaijer, and A. Welbers, “Matching properties of MOS transistors,” *IEEE Journal of Solid-State Circuits*, vol. 24, no. 5, pp. 1433–1439, 1989.

- [33] J. Bastos, M. Steyaert, R. Roovers, P. Kinget, W. Sansen, B. Graindourze, A. Pergoot, and E. Janssens, "Mismatch characterization of small size MOS transistors," in *Proceedings of the International Conference on Microelectronic Test Structures (ICMTS)*, pp. 271–276, 1995.
- [34] J. Croon, M. Rosmeulen, S. Decoutere, W. Sansen, and H. Maes, "An easy-to-use mismatch model for the MOS transistor," *IEEE Journal of Solid-State Circuits*, vol. 37, no. 8, pp. 1056–1064, 2002.
- [35] P. Drennan and C. McAndrew, "Understanding MOSFET mismatch for analog design," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 3, pp. 450–456, 2003.
- [36] T. Serrano-Gotarredona and B. Linares-Barranco, "Systematic width-and-length dependent CMOS transistor mismatch characterization and simulation," *Analog Integrated Circuits and Signal Processing*, vol. 21, no. 3, pp. 271–296, 1999.
- [37] Electronics Research Laboratory of the University of California at Berkeley, "Spice." <http://bwrc.eecs.berkeley.edu/Classes/icbook/SPICE/>.
- [38] Mentor Graphics, "Eldo." <http://www.mentor.com/>.
- [39] Cadence, "Spectre." <http://www.cadence.com/>.
- [40] S. Nassif, A. Strojwas, and S. Director, "A methodology for worst-case analysis of integrated circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 5, no. 1, pp. 104–113, 1986.
- [41] G. Rappitsch, E. Seebacher, M. Kocher, and E. Stadlober, "Spice modeling of process variation using location depth corner models," *IEEE Transactions on Semiconductor Manufacturing*, vol. 17, no. 2, pp. 201–213, 2004.
- [42] M. Sengupta, S. Saxena, L. Daldoss, G. Kramer, S. Minehane, and J. Cheng, "Application specific worst case corners using response surfaces and statistical models," in *Proceedings of the 5th International Symposium on Quality Electronic Design*, pp. 351–356, 2004.
- [43] A. Graupner, W. Schwarz, and R. Schuffny, "Statistical analysis of analog structures through variance calculation," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 49, no. 8, pp. 1071–1078, 2002.
- [44] J. Lei, P. Lima-Filho, M. Styblinski, and C. Singh, "Propagation of variance using a new approximation in system design of integrated circuits," in *Proceedings of the IEEE 1998 National Aerospace and Electronics Conference (NAECON)*, pp. 242–246, 1998.

- [45] B. Spence, L. Tweedie, H. Dawkes, and H. Su, *Visualization for functional design*. Washington, DC, USA: IEEE Computer Society, 1995.
- [46] G. Gielen, P. Wambacq, and W. Sansen, "Symbolic analysis methods and applications for analog circuits: a tutorial overview," *Proceedings of the IEEE*, vol. 82, no. 2, pp. 287–304, 1994.
- [47] J.-Q. Lu, K. Ogawa, T. Adachi, and A. J. Strojwas, "Stochastic interpolation model scheme for statistical circuit design," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, vol. 1, pp. 125–128, 1994.
- [48] G. Yu and P. Li, "Yield-aware analog integrated circuit optimization using geostatistics motivated performance modeling," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 464–469, 2007.
- [49] R. Pratap, P. Sen, C. Davis, R. Mukhopdhyay, G. May, and J. Laskar, "Neurogenetic design centering," *IEEE Transactions on Semiconductor Manufacturing*, vol. 19, no. 2, pp. 173–182, 2006.
- [50] M. Styblinski and S. Aftab, "Combination of interpolation and self-organizing approximation techniques-a new approach to circuit performance modeling," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 12, no. 11, pp. 1775–1785, 1993.
- [51] S. Director and G. Hachtel, "The simplicial approximation approach to design centering," *IEEE Transactions on Circuits and Systems*, vol. 24, no. 7, pp. 363–372, 1977.
- [52] S. Sapatnekar, P. Vaidya, and S. Kang, "Feasible region approximation using convex polytopes," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1786–1789, 1993.
- [53] H. Abdel-Malek and A.-K. Hassan, "The ellipsoidal technique for design centering and region approximation," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 10, no. 8, pp. 1006–1014, 1991.
- [54] J. Wojciechowski and J. Vlach, "Ellipsoidal method for design centering and yield estimation," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 12, no. 10, pp. 1570–1579, 1993.
- [55] J. Ogrodzki and M. Styblinski, "Optimal tolerancing, centering and yield optimization by one-dimensional orthogonal search (ODOS) technique," in *Proceedings of the European Conference on Circuit Theory and Design (ECCTD)*, 1980.

- [56] K. Tahim and R. Spence, “A radial exploration approach to manufacturing yield estimation and design centering,” *IEEE Transactions on Circuits and Systems*, vol. 26, no. 9, pp. 768–774, 1979.
- [57] S. Director, P. Feldmann, and K. Krishna, “Statistical integrated circuit design,” *IEEE Journal of Solid-State Circuits*, vol. 28, no. 3, pp. 193–202, 1993.
- [58] M. Keramat and R. Kielbasa, “Parametric yield optimization of electronic circuits via improved centers of gravity algorithm,” in *Proceedings of the 40th Midwest Symposium on Circuits and Systems*, vol. 2, pp. 1415–1418, 1997.
- [59] X. Li, J. Le, P. Gopalakrishnan, and L. T. Pileggi, “Asymptotic probability extraction for nonnormal performance distributions,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, no. 1, pp. 16–37, 2007.
- [60] E. J. Gumbel, *Statistics of extremes*. Columbia University Press, 1958.
- [61] M. Piera-Martínez, *Modélisation des comportements extrêmes en ingénierie*. PhD thesis, Université Paris Sud Orsay, 2008.
- [62] R. Aitken and S. Idgunji, “Worst-case design and margin for embedded SRAM,” in *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition (DATE)*, pp. 1–6, 2007.
- [63] S. Mukhopadhyay, “A generic method for variability analysis of nanoscale circuits,” in *Proceedings of the IEEE International Conference on Integrated Circuit Design and Technology (ICICDT)*, pp. 285–288, 2008.
- [64] A. Singhee and R. Rutenbar, “Statistical blockade: very fast statistical simulation and modeling of rare circuit events and its application to memory design,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, no. 8, pp. 1176–1189, 2009.
- [65] G. Gielen and R. Rutenbar, “Computer-aided design of analog and mixed-signal integrated circuits,” *Proceedings of the IEEE*, vol. 88, no. 12, pp. 1825–1854, 2000.
- [66] K. Singhal and J. Pinel, “Statistical design centering and tolerancing using parametric sampling,” *IEEE Transactions on Circuits and Systems*, vol. 28, no. 7, pp. 692–702, 1981.
- [67] D. E. Hocevar, M. R. Lightner, and T. N. Trick, “An extrapolated yield approximation technique for use in yield maximization,” vol. 3, no. 4, pp. 279–287, 1984.

- [68] F. Schenkel, M. Pronath, S. Zizala, R. Schwencker, H. Graeb, and K. Antreich, "Mismatch analysis and direct yield optimization by spec-wise linearization and feasibility-guided search," in *Proceedings of the 38th Design Automation Conference (DAC)*, pp. 858–863, 2001.
- [69] K. Antreich and R. Koblitz, "Design centering by yield prediction," *IEEE Transactions on Circuits and Systems*, vol. 29, no. 2, pp. 88–96, 1982.
- [70] M. Styblinski and A. Ruszczynski, "Stochastic approximation approach to statistical circuit design," *Electronics Letters*, vol. 19, no. 8, pp. 300–302, 1983.
- [71] Z. Wang and S. Director, "An efficient yield optimization method using a two step linear approximation of circuit performance," in *Proceedings of the European Design and Test Conference (EDAC)*, pp. 567–571, 1994.
- [72] R. S. Soin and R. Spence, "Statistical exploration approach to design centring," *IEE Proceedings G Electronic Circuits and Systems*, vol. 127, no. 6, pp. 260–269, 1980.
- [73] S. Director, P. Feldmann, and K. Krishna, "Optimization of parametric yield: A tutorial," in *Proceedings of the IEEE 1992 Custom Integrated Circuits Conference*, pp. 3.1.1–3.1.8, 1992.
- [74] A. Seifi, K. Ponnambalam, and J. Vlach, "A unified approach to statistical design centering of integrated circuits with correlated parameters," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 46, no. 1, pp. 190–196, 1999.
- [75] K. Antreich, H. Graeb, and C. Wieser, "Circuit analysis and optimization driven by worst-case distances," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 13, no. 1, pp. 57–71, 1994.
- [76] K. Krishna and S. Director, "The linearized performance penalty (LPP) method for optimization of parametric yield and its reliability," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 14, no. 12, pp. 1557–1568, 1995.
- [77] R. Schwencker, F. Schenkel, M. Pronath, and H. Graeb, "Analog circuit sizing using adaptive worst-case parameter sets," in *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition (DATE)*, pp. 581–585, 2002.
- [78] A. Ben-Tal and A. Nemirovski, "Robust convex optimization," *Mathematics of Operations Research*, vol. 23, pp. 769–805, 1998.

- [79] T. Mukherjee, L. Carley, and R. Rutenbar, “Efficient handling of operating range and manufacturing line variations in analog cell synthesis,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 19, no. 8, pp. 825–839, 2000.
- [80] Y. Xu, X. Li, K.-L. Hsiung, S. Boyd, and I. Nausieda, “OPERA: optimization with ellipsoidal uncertainty for robust analog IC design,” in *Proceedings of the 42nd Design Automation Conference (DAC)*, pp. 632–637, 2005.
- [81] K.-L. Hsiung, S.-J. Kim, and S. Boyd, “Tractable approximate robust geometric programming,” *Optimization and Engineering*, vol. 9, June 2008.
- [82] X. Li, P. Gopalakrishnan, Y. Xu, and L. T. Pileggi, “Robust analog/RF circuit design with projection-based performance modeling,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, no. 1, pp. 2–15, 2007.
- [83] A. Dharchoudhury and S. Kang, “Worst-case analysis and optimization of VLSI circuit performances,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 14, pp. 481–492, April 1995.
- [84] A. Lemke, L. Hedrich, and E. Barke, “Analog circuit sizing based on formal methods using affine arithmetic,” in *Proceedings of the IEEE/ACM International Conference on Computer Aided Design (ICCAD)*, pp. 486–489, 2002.
- [85] S. Aftab and M. Styblinski, “IC variability minimization using a new C_p and C_{pk} based variability/performance measure,” in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, vol. 1, pp. 149–152, 1994.
- [86] G. Debyser and G. Gielen, “Efficient analog circuit synthesis with simultaneous yield and robustness optimization,” in *Proceedings of the 1998 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 308–311, 1998.
- [87] G. Gielen, “Design methodologies and tools for circuit design in CMOS nanometer technologies,” in *Proceeding of the 36th European Solid-State Device Research Conference (ESSDERC)*, pp. 21–32, 2006.
- [88] J. Kleijnen, *Statistical tools for simulation practitioners*. Marcel Dekker, Inc., 1986.
- [89] A. Jourdan, “Planification d’expériences numériques,” *La revue de MODULAD*, vol. 33, pp. 63–73, Juin 2005.

- [90] R. Jin, W. Chen, and T. W. Simpson, "Comparative studies of metamodelling techniques under multiple modelling criteria," *Structural and Multidisciplinary Optimization*, vol. 23, pp. 1–13, December 2001.
- [91] J. Sacks, W. Welch, T. Mitchell, and H. Wynn, "Designs and analysis of computer experiments," *Statistical Science*, vol. 4, pp. 409–435, 1989.
- [92] G. G. Wang and S. Shan, "Review of metamodeling techniques in support of engineering design optimization," *Journal of Mechanical Design*, vol. 129, no. 4, pp. 370–380.
- [93] A. Singhee and R. Rutenbar, "Beyond low-order statistical response surfaces: latent variable regression for efficient, highly nonlinear fitting," in *Proceedings of the 44th ACM/IEEE Design Automation Conference (DAC)*, pp. 256–261, 2007.
- [94] W. Daems, G. Gielen, and W. Sansen, "Simulation-based generation of posynomial performance models for the sizing of analog integrated circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, no. 5, pp. 517–534, 2003.
- [95] M. M. Hershenson, S. Boyd, and T. Lee, "Optimal design of a CMOS op-amp via geometric programming," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 20, no. 1, pp. 1–21, 2001.
- [96] T. Kiely and G. Gielen, "Performance modeling of analog integrated circuits using least-squares support vector machines," in *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition (DATE)*, p. 10448, IEEE Computer Society, 2004.
- [97] J. H. Friedman and W. Stuetzle, "Projection pursuit regression," *Journal of the American Statistical Association*, vol. 76, pp. 817–823, 1981.
- [98] X. Li and Y. Cao, "Projection-based piecewise-linear response surface modeling for strongly nonlinear VLSI performance variations," in *Proceedings of the 9th International Symposium on Quality Electronic Design (ISQED)*, pp. 108–113, 2008.
- [99] R. Ghanem and P. Spanos, *Stochastic Finite Elements: A Spectral Approach*. Springer Verlag, New York, 1991.
- [100] X. Li and H. Liu, "Statistical regression for efficient high-dimensional modeling of analog and mixed-signal performance variations," in *Proceedings of the 45th ACM/IEEE Design Automation Conference (DAC)*, pp. 38–43, 2008.

- [101] N. Dong and J. Roychowdhury, “General-purpose nonlinear model-order reduction using piecewise-polynomial representations,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 2, pp. 249–264, 2008.
- [102] P. Feldmann and R. W. Freund, “Reduced-order modeling of large linear subcircuits via a block Lanczos algorithm,” in *Proceedings of the 32nd annual ACM/IEEE Design Automation Conference (DAC)*, (New York, NY, USA), pp. 474–479, ACM, 1995.
- [103] J. Goupy, *Introduction aux plans d’expériences*. Dunod, 2006.
- [104] M.Sado and G.Sado, *Les plans d’expériences*. AFNOR, 2000.
- [105] R. Fisher, *The design of experiments*. Oliver and Boyd, 1935.
- [106] G. E. P. Box and N. R. Draper, *Empirical Model-building and Response Surfaces*. Wiley, 1987.
- [107] G. Taguchi and D. Clausing, “Robust quality,” *Harvard Business Review*, pp. 65–75, January-February 1990.
- [108] J. Goupy, “Les plans d’expériences,” *La revue de MODULAD*, vol. 34, pp. 74–116, Juillet 2006.
- [109] Mathworks, “Matlab.” <http://www.mathworks.fr/products/matlab/>.
- [110] Stat-Ease, “Design-Expert.” <http://www.statease.com/>.
- [111] R. Plackett and J. P. Burman, “The design of optimum multifactorial experiments,” *Biometrika*, vol. 33, no. 4, pp. 305–325, 1946.
- [112] G. E. P. Box and K. B. Wilson, “On the experimental attainment of optimum conditions,” *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 13, no. 1, pp. 1–45, 1951.
- [113] K. Low and S. Director, “An efficient methodology for building macromodels of IC fabrication processes,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 8, no. 12, pp. 1299–1313, 1989.
- [114] S. Jaschke, “The Cornish-Fisher-Expansion in the context of Delta - Gamma - Normal approximations,” Tech. Rep. 2001-54, Humboldt Universität Berlin.
- [115] A. M. Mathai and S. B. Provost, *Quadratic Forms in Random Variables: Theory and Applications*. M. Dekker, 1992.

- [116] E. A. Cornish and R. A. Fisher, “Moments and cumulants in the specification of distributions,” *Revue de l’Institut International de Statistique*, vol. 5, pp. 307–320, 1937.
- [117] P. Zangari, “A VaR methodology for portfolios that include options,” *RiskMetrics Monitor*, pp. 4–12, January 1996.
- [118] N. L. Johnson, S. Kotz, and N. Balakrishnan, *Continuous Univariate Distributions*, vol. 1. Wiley, 1994.
- [119] STMicroelectronics. <http://www.st.com>.
- [120] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter, *Applied Interval Analysis, with Examples in Parameter and State Estimation, Robust Control and Robotics*. London: Springer-Verlag, 2001.
- [121] K. Swings and W. Sansen, “ARIADNE: A constraint-based approach to computer-aided synthesis and modeling of analog integrated circuits,” *Analog Integrated Circuits and Signal Processing*, vol. 3, pp. 197–215, May 1993.
- [122] J. Michel and F. Schwartz, “Analogue circuit sizing method using interval analysis,” in *Proceedings of the joint 6th International IEEE Northeast Workshop on Circuits and Systems and TAISA Conference (NEWCAS-TAISA)*, pp. 331–334, 2008.
- [123] R. E. Moore, *Interval Analysis*. Prentice-Hall, 1966.
- [124] E. Hansen and G. W. Walster, *Global Optimization Using Interval Analysis*. 2003.
- [125] D. Ratz and T. Csendes, “On the selection of subdivision directions in interval branch-and-bound methods for global optimization,” *Journal of Global Optimization*, vol. 7, pp. 183–207, 1995.
- [126] A. Neumaier, *Interval Methods for Systems of Equations*. Cambridge University Press, 1990.
- [127] F. Benhamou, F. Goualard, L. Granvilliers, and J.-F. Puget, “Revising hull and box consistency,” in *International Conference on Logic Programming*, pp. 230–244, 1999.
- [128] L. Jaulin, X. Baguenard, and M. Dao, “Interval peeler.” <http://www.ensieta.fr/jaulin/demo.html>.
- [129] A. d’Aspremont and S. Boyd, “Relaxations and randomized methods for nonconvex QCQPs,” tech. rep., Stanford University, 2003.
- [130] F. Roupin, “L’approche par programmation semidéfinie en optimisation combinatoire,” *Bulletin de la ROADEF*, no. 13, pp. 7–11, 2004.

- [131] S. Rump, “INTLAB - INTerval LABoratory,” in *Developments in Reliable Computing* (T. Csendes, ed.), pp. 77–104, Dordrecht: Kluwer Academic Publishers, 1999. <http://www.ti3.tu-harburg.de/rump/>.
- [132] F. Messine, Méthodes d’Optimisation Globale basées sur l’Analyse d’Intervalle pour la Résolution des Problèmes avec Contraintes. PhD thesis, Institut National Polytechnique de Toulouse, 1997.
- [133] G. Stehr, M. Pronath, F. Schenkel, H. Graeb, and K. Antreich, “Initial sizing of analog integrated circuits by centering within topology-given implicit specifications,” in *Proceedings of the International Conference on Computer Aided Design (ICCAD)*, pp. 241–246, 2003.
- [134] P. Allen and D. Holberg, *CMOS Analog Circuit Design, second edition*. Oxford University Press, 2002.
- [135] F. Silveira, D. Flandre, and P. Jespers, “A g_m/i_d based methodology for the design of CMOS analog circuits and its application to the synthesis of a silicon-on-insulator micropower OTA,” *IEEE Journal of Solid-State Circuits*, vol. 31, September 1996.
- [136] A. Girardi and S. Bampi, “Power constrained design optimization of analog circuits based on physical g_m/I_D characteristics,” in *Proceedings of the 19th annual symposium on Integrated Circuits and Systems Design*, pp. 89–93, 2006.
- [137] L. Wang, S. Shan, and G. Wang, “Mode-pursuing sampling method for global optimization on expensive black-box functions,” *Journal of Engineering Optimization*, vol. 36, pp. 419–438, August 2004.
- [138] F. Tissafi-Drissi, *Méthodes et outils de synthèse pour systèmes multi-domaine*. PhD thesis, Ecole Centrale de Lyon, 2004.
- [139] F. Tissafi-Drissi, I. O’Connor, and F. Gaffiot, “RUNE: Platform for automated design of integrated multi-domain systems. Application to high-speed CMOS photoreceiver front-ends,” in *Proceedings of the conference on Design, Automation and Test in Europe (DATE)*, p. 16–21, 2004.

Publications

Conférences internationales

H. Filiol, I. O'Connor, and D. Morche, "A new approach for variability analysis of analog ICs," in *Proceedings of the IEEE North-East Workshop on Circuits and Systems and TAISA Conference*, pp. 1-4, Toulouse, France, 2009.

H. Filiol, I. O'Connor, and D. Morche, "Piecewise-polynomial modeling for analog circuit performance metrics," in *Proceedings of the European Conference on Circuit Theory and Design (ECTD)*, pp. 237-240, Antalya, Turquie, 2009.

Conférences francophones

H. Filiol, I. O'Connor, and D. Morche, "Méthodes de conception analogique robuste aux dispersions paramétriques", *11ème Journées Nationales du Réseau Doctoral en Microélectronique (JNRDM)*, Bordeaux, France, 2008.

H. Filiol, I. O'Connor, and D. Morche, "Méthode de conception analogique robuste aux dispersions paramétriques", *Colloque national du GDR SOC-SIP*, Paris, France, 2008.

H. Filiol, I. O'Connor, and D. Morche, "Méthode de conception analogique robuste aux dispersions paramétriques", *Ecole d'hiver Francophone sur les Technologies de Conception des systèmes embarqués Hétérogènes (FETCH)*, Chexbres, Suisse, 2009.

H. Filiol, I. O'Connor, and D. Morche, "A new approach for efficient variability analysis at transistor level in advanced CMOS technologies", *Colloque national du GDR SOC-SIP*, Paris, France, 2009.

H. Filiol, I. O'Connor, and D. Morche, "Variability analysis of ICs using the Cornish-Fisher approximation," *VARI2010*, Montpellier, France, 2010.

Annexe A : Éléments de probabilité

1. Variable aléatoire

Variable aléatoire réelle. De manière informelle, une variable aléatoire réelle peut être vue comme une quantité réelle qui prend des valeurs de façon imprévisible mais néanmoins quantifiable sous la forme d'une probabilité ; elle permet de modéliser le résultat d'un phénomène non-déterministe. Une variable aléatoire peut être caractérisée de façon équivalente par sa densité de probabilité, sa fonction de répartition, ses moments ou sa fonction caractéristique.

Densité de probabilité. La densité de probabilité f_X d'une variable aléatoire X est une fonction positive qui vérifie :

$$\int_{-\infty}^{+\infty} f_X(x) dx = 1 \quad (\text{A.1})$$

Elle permet de définir la probabilité que la variable aléatoire X prenne une valeur appartenant à l'intervalle $[a, b[$:

$$P(a \leq X < b) = \int_a^b f_X(x) dx \quad (\text{A.2})$$

Fonction de répartition. La fonction de répartition F_X d'une variable aléatoire X est une fonction de $\Re$ dans $[0, 1]$ qui vérifie :

$$F_X(x) = P(X < x) \quad (\text{A.3})$$

F_X est la primitive de f_X qui vaut 0 en $-\infty$. Elle possède les propriétés suivantes :

$$F_X(x) \text{ est croissante, } \lim_{x \rightarrow -\infty} F_X(x) = 0, \quad \lim_{x \rightarrow +\infty} F_X(x) = 1 \quad (\text{A.4})$$

Quantile, centile (ou percentile). Soit F_X la fonction de répartition d'une variable aléatoire X et α un nombre compris entre 0 et 1. Le quantile d'ordre α , noté q_α , de la distribution de X est le point qui vérifie (cf. Figure 7.1) :

$$F_X(q_\alpha) = \alpha \Leftrightarrow q_\alpha = F_X^{-1}(\alpha) \quad (\text{A.5})$$

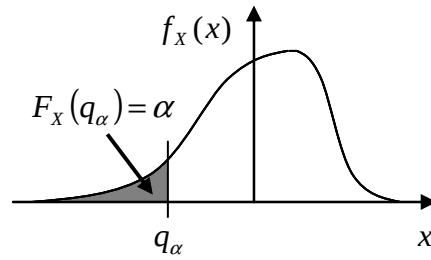


Figure 7.1 : Quantile d'ordre α de la distribution de X

Lorsque q_α est exprimé en pourcents, on parle alors de centile ou de percentile.

Moyenne, espérance. Soit X une variable aléatoire dont la densité de probabilité est notée f_X . La moyenne, également appelée espérance, de X est définie par :

$$\mu = E(X) = \int_{-\infty}^{+\infty} x f_X(x) dx \quad (\text{A.6})$$

La moyenne est un indicateur de la tendance centrale de la distribution de X .

Soient X et Y deux variables aléatoires et $a \in \mathfrak{R}$, les règles de calcul pour la moyenne sont les suivantes :

$$\begin{cases} E(a) = a \\ E(aX) = aE(X) \\ E(X + Y) = E(X) + E(Y) \\ E(XY) = E(X)E(Y) \text{ si } X \text{ et } Y \text{ sont indépendantes} \end{cases} \quad (\text{A.7})$$

Moments. La généralisation de la moyenne conduit à la notion de moment. Le moment d'ordre $n \geq 0$ d'une variable aléatoire X est donné par :

$$E(X^n) = \int_{-\infty}^{+\infty} x^n f_X(x) dx \quad (\text{A.8})$$

Le moment d'ordre 1 correspond donc à la moyenne de X .

Il est parfois plus aisé d'utiliser les moments centrés d'une variable aléatoire X . Son moment centré d'ordre n est défini par :

$$\mu_n = E[(X - \mu)^n] = \int_{-\infty}^{+\infty} (x - \mu)^n f_X(x) dx \quad (\text{A.9})$$

où μ est la moyenne de X .

Variance, écart type. Le moment centré d'ordre 2 d'une variable aléatoire X est appelée variance :

$$V(X) = \mu_2 = E[(X - \mu)^2] = E(X^2) - \mu^2 \quad (\text{A.10})$$

La racine carrée de la variance de X est son écart type :

$$\sigma = \sqrt{V(X)} \quad (\text{A.11})$$

La variance et l'écart type renseignent sur la dispersion des valeurs de X autour de la moyenne.

Soient X et Y deux variables aléatoires et $a \in \mathfrak{R}$. Les règles de calcul pour la variance sont les suivantes :

$$\begin{cases} V(a) = 0 \\ V(aX) = a^2 V(X) \\ V(X + Y) = V(X) + V(Y) + 2 \text{cov}(X, Y) \end{cases} \quad (\text{A.12})$$

où $\text{cov}(X, Y)$ est la covariance de X et de Y qui est égale à $E[(X - E(X))(Y - E(Y))]$.

Variable centrée réduite. Si X une variable aléatoire de moyenne μ et d'écart type σ , on appelle alors variable centrée réduite la variable définie par :

$$Y = \frac{X - \mu}{\sigma} \quad (\text{A.13})$$

La moyenne et l'écart type de Y valent respectivement 0 et 1.

Fonction caractéristique. La fonction caractéristique d'une variable aléatoire X est la transformée de Fourier de sa densité de probabilité f_X :

$$\varphi_X(t) = \int_{-\infty}^{+\infty} e^{jtx} f_X(x) dx = E(e^{jtx}) \quad (\text{A.14})$$

L'intérêt de la fonction caractéristique est de permettre un calcul aisé des moments de X . En effet, le développement de la fonction caractéristique de X en série de MacLaurin donne :

$$\varphi_X(t) = \sum_{n=0}^{\infty} E(X^n) \frac{(jt)^n}{n!} \quad (\text{A.15})$$

où $E(X^n)$ sont les moments d'ordre n de X . Les moments peuvent donc être obtenus à partir de la formule :

$$E(X^n) = (-j)^n \left[\frac{d^n \varphi_X(t)}{dt^n} \right]_{t=0} \quad (\text{A.16})$$

Cumulants. Les cumulants d'ordre n d'une variable aléatoire X sont obtenus à partir du développement en série de MacLaurin du logarithme de la fonction caractéristique :

$$\ln(\varphi_X(t)) = \sum_{n=1}^{\infty} \kappa_n \frac{(jt)^n}{n!} \quad (\text{A.17})$$

où κ_n est le cumulant d'ordre n . La formule des cumulants est donc la suivante :

$$\kappa_n = (-j)^n \left[\frac{d^n \ln \varphi_X(t)}{dt^n} \right]_{t=0} \quad (\text{A.18})$$

Dans la mesure où les cumulants sont générés à partir de la fonction caractéristique comme les moments, ils sont conceptuellement similaires à ces derniers. Une relation existe donc entre les cumulants et les moments centrés μ_n d'une variable aléatoire :

$$\begin{aligned} \kappa_1 &= \mu \\ \kappa_2 &= \mu_2 \\ \kappa_3 &= \mu_3 \\ \kappa_4 &= \mu_4 - 3\mu_2^2 \\ \kappa_5 &= \mu_5 - 10\mu_2\mu_3 \end{aligned} \quad (\text{A.19})$$

Les deux premiers cumulants correspondent respectivement à la moyenne et à la variance.

2. Vecteur de variables aléatoires

Vecteur aléatoire. Un vecteur aléatoire est un vecteur composé de k variables aléatoires $X_1, X_2, \dots, X_k$:

$$\mathbf{X} = \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_k \end{pmatrix} \quad (\text{A.20})$$

Fonction de répartition conjointe, densité de probabilité conjointe. La fonction de répartition F_X et la densité de probabilité f_X d'un vecteur aléatoire $\mathbf{X}$ sont respectivement définies par :

$$F_X(x_1, x_2, \dots, x_k) = P[(X_1 \leq x_1) \cap (X_2 \leq x_2) \cap \dots \cap (X_k \leq x_k)] \quad (\text{A.21})$$

$$f_X(x_1, x_2, \dots, x_k) = \frac{\partial^k F_X}{\partial x_1 \partial x_2 \dots \partial x_k}(x_1, x_2, \dots, x_k) \quad (\text{A.22})$$

Densité de probabilité marginale. La densité de probabilité marginale selon X_i est obtenue en considérant la variable aléatoire X_i indépendamment des autres composantes du vecteur $\mathbf{X}$:

$$f_{X_i}(x_i) = \int_{x_1=-\infty}^{+\infty} \dots \int_{x_{i-1}=-\infty}^{+\infty} \int_{x_{i+1}=-\infty}^{+\infty} \dots \int_{x_k=-\infty}^{+\infty} f_X(x_1, \dots, x_i, \dots, x_k) dx_1 \dots dx_{i-1} dx_{i+1} \dots dx_k \quad (\text{A.23})$$

Moyenne. Soit $\mathbf{X}$ est un vecteur aléatoire de dimension k . La moyenne de $\mathbf{X}$ est un vecteur dont les composantes sont les moyennes de chaque variable X_i obtenues avec les densités de probabilité marginales selon X_i :

$$E(\mathbf{X}) = \begin{pmatrix} E(X_1) \\ E(X_2) \\ \vdots \\ E(X_k) \end{pmatrix} \quad (\text{A.24})$$

Matrice de covariance. La matrice de covariance d'un vecteur aléatoire $\mathbf{X}$ est une matrice carrée définie par :

$$\Sigma = E[(\mathbf{X} - \mu)(\mathbf{X} - \mu)^T] \quad (\text{A.25})$$

où μ est la moyenne du vecteur $\mathbf{X}$. Les termes diagonaux de Σ correspondent aux variances des composantes X_i , tandis que les termes hors de la diagonale représentent leurs covariances.

Forme linéaire. Soit Y une variable aléatoire définie par $Y = \mathbf{B}^T \mathbf{X} + a$ où $\mathbf{B} \in \Re^k$, $a \in \Re$ et $\mathbf{X}$ est un vecteur aléatoire de dimension k . La moyenne et la variance de Y sont données par :

$$\begin{cases} E(Y) = \mathbf{B}^T E(\mathbf{X}) + a \\ V(Y) = \mathbf{B}^T \Sigma \mathbf{B} \end{cases} \quad (\text{A.26})$$

3. Loi normale

Densité de probabilité. La densité de probabilité d'une variable aléatoire X distribuée selon une loi normale est donnée par :

$$f_X(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \quad (\text{A.27})$$

où μ et σ représentent respectivement la moyenne et l'écart type de X . Une loi normale est entièrement définie par sa moyenne et son écart type ; elle sera notée $N(\mu, \sigma)$. La loi $N(0, 1)$ est appelée loi normale centrée réduite.

Somme de variables normales indépendantes. Si $X_i \sim N(\mu_i, \sigma_i)$ est une suite de n variables aléatoires normales indépendantes, alors la variable $Y = \sum_{i=1}^n X_i$ est également une variable normale de moyenne $\mu = \sum_{i=1}^n \mu_i$ et de variance $\sigma^2 = \sum_{i=1}^n \sigma_i^2$.

4. Loi du Khi-2 non centrée

Soit $X_i \sim N(\mu_i, \sigma_i)$ une suite de n variables aléatoires normales indépendantes, alors la variable $Y = \sum_{i=1}^n \left(\frac{X_i}{\sigma_i} \right)^2$ est distribuée selon une loi du Khi-2 non centrée. Deux paramètres caractérisent cette loi : le nombre de degrés de liberté n , et le paramètre de non-centralité $\lambda = \sum_{i=1}^n \left(\frac{\mu_i}{\sigma_i} \right)^2$. Une loi du Khi-2 non centrée sera donc notée $\chi^2(n, \lambda)$.

La fonction caractéristique d'une loi Khi-2 non centrée est donnée par :

$$\varphi(t) = (1 - 2jt)^{-\frac{n}{2}} \exp\left(\frac{jt\lambda}{1 - 2jt}\right) \quad (\text{A.28})$$

Annexe B : Problème d'optimisation

Tout problème d'optimisation peut se formuler en faisant intervenir trois éléments distincts :

- la fonction de coût qui modélise la grandeur que l'on cherche à minimiser,
- les variables de décision sur lesquelles il est possible d'agir afin de minimiser la fonction de coût,
- les contraintes qui définissent les valeurs que les variables de décision peuvent prendre, c'est-à-dire le domaine de faisabilité.

Un problème d'optimisation peut donc s'écrire sous la forme générale suivante :

$$\begin{array}{ll} \text{minimiser} & f(X) \\ \text{tel que} & g_l(X) < 0, \quad l \in \{1, \dots, p\} \\ & X \in D \end{array} \quad (\text{B.1})$$

où X est le vecteur des variables de décision, $f: \Re^k \mapsto \Re$ est la fonction de coût, tandis que $g_l: \Re^k \mapsto \Re$ pour l allant de 1 à p et $D \subset \Re^k$ représentent les contraintes. Le domaine de faisabilité du problème d'optimisation (B.1) est défini par :

$$C = \{X \in D \mid g_l(X) < 0, l \in \{1, \dots, p\}\} \quad (\text{B.2})$$

Ainsi, résoudre le problème d'optimisation (B.1) consiste à trouver un $X^* \in C$ tel que $f(X^*) \leq f(X)$ pour tout $X \in C$. S'il n'y a pas de contraintes, on parle de problème d'optimisation sans contraintes. Par ailleurs, suivant la nature de la fonction de coût et des contraintes (linéaire, quadratique, convexe, etc.), les méthodes numériques de résolution des problèmes d'optimisation sont appelées programmation linéaire, quadratique, convexe, etc. La non-convexité signifie que le problème peut avoir plusieurs optima locaux ; toute la difficulté consiste donc à déterminer l'optimum global (s'il existe), on parle alors d'optimisation globale par opposition à l'optimisation locale.

Annexe C : Arithmétique par intervalles

1. Définitions

Intervalle. Un intervalle réel compact, noté $[X]$, est un sous-ensemble connexe de $\Re$. Il est caractérisé par sa borne inférieure $\underline{X} \in \Re$ et sa borne supérieure $\overline{X} \in \Re$:

$$[X] = [\underline{X}, \overline{X}] = \{x \in \Re \mid \underline{X} \leq x \leq \overline{X}\} \quad (\text{C.1})$$

L'ensemble des intervalles compacts réels est noté IR .

La largeur d'un intervalle est définie par :

$$w([X]) = \overline{X} - \underline{X} \quad (\text{C.2})$$

Vecteur d'intervalles ou pavés. Un vecteur d'intervalles, noté $[X]$, est un sous-ensemble connexe de $\Re^k$. Il est défini par le produit cartésien de k intervalles :

$$[X] = [X_1] \times [X_2] \times \dots \times [X_k] \quad (\text{C.3})$$

La largeur d'un vecteur d'intervalles est donnée par :

$$w([X]) = \max_{1 \leq i \leq k} w([X_i]) \quad (\text{C.4})$$

Les vecteurs d'intervalles sont communément appelés pavés.

Opérations usuelles. Les opérations arithmétiques usuelles peuvent être étendues aux intervalles en raisonnant sur leurs bornes :

$$\begin{aligned}
[\underline{X}, \overline{X}] + [\underline{Y}, \overline{Y}] &= [\underline{X} + \underline{Y}, \overline{X} + \overline{Y}], \\
[\underline{X}, \overline{X}] - [\underline{Y}, \overline{Y}] &= [\underline{X} - \underline{Y}, \overline{X} - \overline{Y}], \\
[\underline{X}, \overline{X}] * [\underline{Y}, \overline{Y}] &= [\min(\underline{X} * \underline{Y}, \underline{X} * \overline{Y}, \overline{X} * \underline{Y}, \overline{X} * \overline{Y}), \max(\underline{X} * \underline{Y}, \underline{X} * \overline{Y}, \overline{X} * \underline{Y}, \overline{X} * \overline{Y})], \\
[\underline{X}, \overline{X}] / [\underline{Y}, \overline{Y}] &= [\min(\underline{X} / \underline{Y}, \underline{X} / \overline{Y}, \overline{X} / \underline{Y}, \overline{X} / \overline{Y}), \max(\underline{X} / \underline{Y}, \underline{X} / \overline{Y}, \overline{X} / \underline{Y}, \overline{X} / \overline{Y})] \text{ avec } 0 \notin [\underline{Y}, \overline{Y}] \\
[\underline{X}, \overline{X}]^2 &= \begin{cases} [\overline{X}^2, \underline{X}^2] & \text{si } \overline{X} \leq 0, \\ [0, \max(\underline{X}^2, \overline{X}^2)] & \text{si } 0 \in [\underline{X}, \overline{X}], \\ [\underline{X}^2, \overline{X}^2] & \text{si } \underline{X} \geq 0, \end{cases} \\
\sqrt{[\underline{X}, \overline{X}]} &= [\sqrt{\underline{X}}, \sqrt{\overline{X}}] \text{ avec } [\underline{X}, \overline{X}] \subset [0, +\infty[,
\end{aligned} \tag{C.5}$$

Fonction d'inclusion. Soit une fonction $f : \Re^k \rightarrow \Re$. L'image d'un vecteur d'intervalles $[X]$ par la fonction f est définie par :

$$f([X]) = \{f(X), X \in [X]\} \tag{C.6}$$

Une fonction $F : IR^k \rightarrow IR$ est appelée fonction d'inclusion pour f si et seulement si :

$$\forall [X] \in IR, \quad f([X]) \subseteq F([X]) \tag{C.7}$$

Les fonctions d'inclusion traitent des intervalles ; elles permettent d'obtenir un encadrement $F([X])$ de l'image $f([X])$. Ainsi, si une valeur n'appartient pas à $F([X])$, alors elle ne peut pas appartenir à $f([X])$ en vertu de la propriété d'inclusion (C.7). Cette propriété est mise à profit dans les algorithmes d'optimisation pour tester l'appartenance du minimum dans un pavé. Les fonctions d'inclusion les plus simples sont obtenues en remplaçant les variables réelles par les intervalles correspondants. Les fonctions d'inclusion construites ainsi sont appelées fonctions d'inclusion naturelles.

Exemple. Soit la fonction f de $[X] = [1, 3] \times [1, 3]$ dans $\Re$ définie par $(x_1, x_2) \mapsto x_1 + 2x_2$. L'image de $[X]$ par la fonction d'inclusion naturelle de f est donnée par :

$$F([X]) = [1, 3] + 2 * [1, 3] = [3, 9] \tag{C.8}$$

2. Surestimation

Le principal inconvénient de l'arithmétique par intervalles est la surestimation. Cette surestimation est causée par deux phénomènes distincts : l'effet enveloppant (« wrapping effect ») et la décorrélation des données.

Effet enveloppant. L'image d'un pavé $[X]$ par une fonction f de $\mathfrak{R}^m$ dans $\mathfrak{R}^n$ est un ensemble de forme quelconque. Or l'encadrement $F([X])$ proposé par la fonction d'inclusion F de f est un pavé qui contient $f([X])$ dont le volume peut être beaucoup plus grand que celui de $f([X])$ comme illustré sur la Figure 7.2.

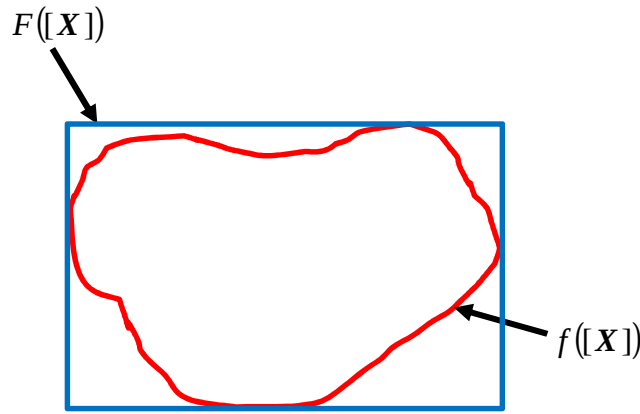


Figure 7.2 : Illustration de l'effet enveloppant

Décorrélation des données. Lorsqu'une variable apparaît plusieurs fois dans une expression, la dépendance entre les intervalles n'est pas prise en compte.

Exemple. Soient deux fonctions f_1 et f_2 définies par $f_1 : x \mapsto (x-1)^2$ et $f_2 : x \mapsto x^2 - 2x + 1$. Ces deux fonctions sont équivalentes en arithmétique réelle, cependant elles ne le sont pas en arithmétique par intervalles. Effet, si on calcule l'image de l'intervalle $[-1, +1]$ par les fonctions d'inclusion F_1 et F_2 , on obtient :

$$\begin{aligned} F_1([-1, +1]) &= ([-1, +1] - 1)^2 = [0, 4] \\ F_2([-1, +1]) &= [-1, +1]^2 - 2 * [-1, +1] + 1 = [-1, +4] \end{aligned} \tag{C.9}$$

L'encadrement fourni par la fonction d'inclusion F_2 est donc plus large que celui fourni par la fonction d'inclusion F_1 (cf. Figure 7.3(a)-(b)).

Théorème de Moore. Si f est une fonction de $\Re^k$ dans $\Re$ définie par une expression où chaque variable x_i apparaît au plus une fois, alors pour tout pavé $[X] \subset \Re^k$, l'évaluation par intervalles obtenue en remplaçant x_i par l'intervalle correspondant est exactement égale à $f([X])$.

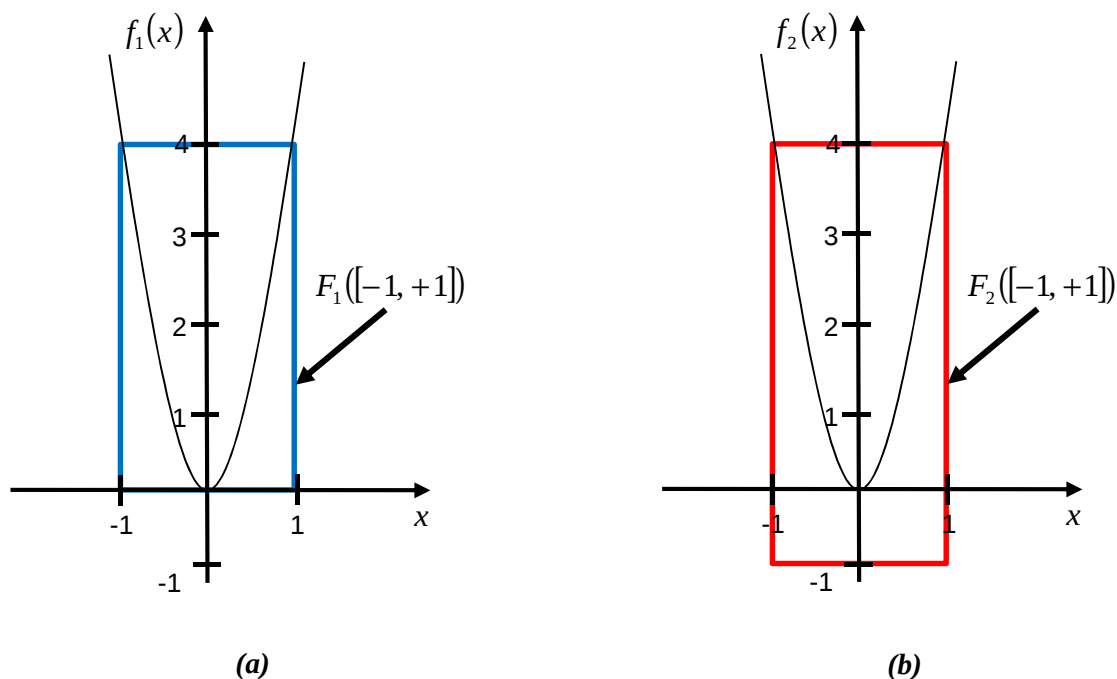
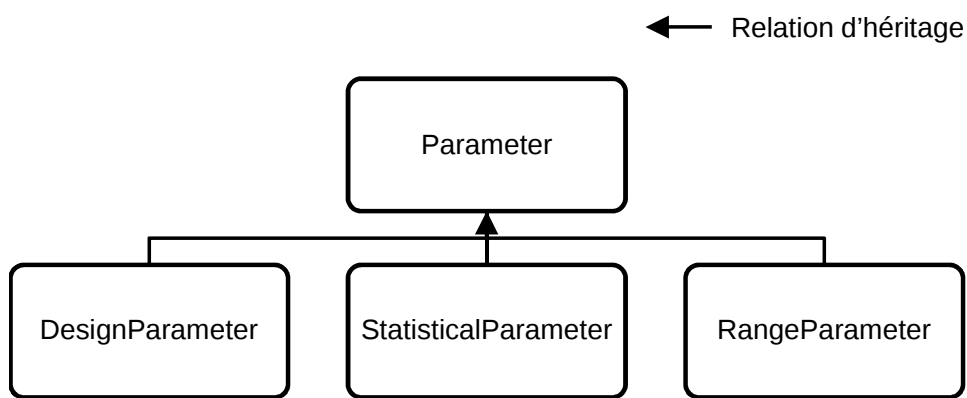


Figure 7.3 : Illustration de l'effet de la décorrélation des données

Annexe D : Implémentation en Java

Les algorithmes de conception robuste par intervalles ont été implémentés en Java afin de s'intégrer à la plateforme de synthèse multi-domaine RUNE [138, 139] déjà développée en Java. L'implémentation s'articule autour des éléments suivants.

Les paramètres. Les différents types de paramètres, à savoir les paramètres de conception, technologiques et environnementaux, sont représentés à partir de la classe mère ***Parameter*** et de ses classes dérivées (cf. Figure 7.4).



*Figure 7.4 : Représentation UML simplifiée de la classe mère **Parameter** et des classes dérivées*

Les fonctions. A partir de la classe mère ***Function***, les différentes représentations possibles d'une fonction sont traitées (cf. Figure 7.5) : les modèles analytiques exprimés au moyen des opérateurs et des fonctions de base (+, -, ×, ÷, exp(), sin(), etc...) , les modèles décrits sous forme de code Java (exemple : estimation d'un quantile à partir des cumulants et du développement de Cornish-Fisher) ou d'un autre langage (exemple : extraction des valeurs des performances analogiques avec un simulateur électrique) et enfin les méta-modèles (exemple : approximation polynomiale construite avec les plans d'expériences).

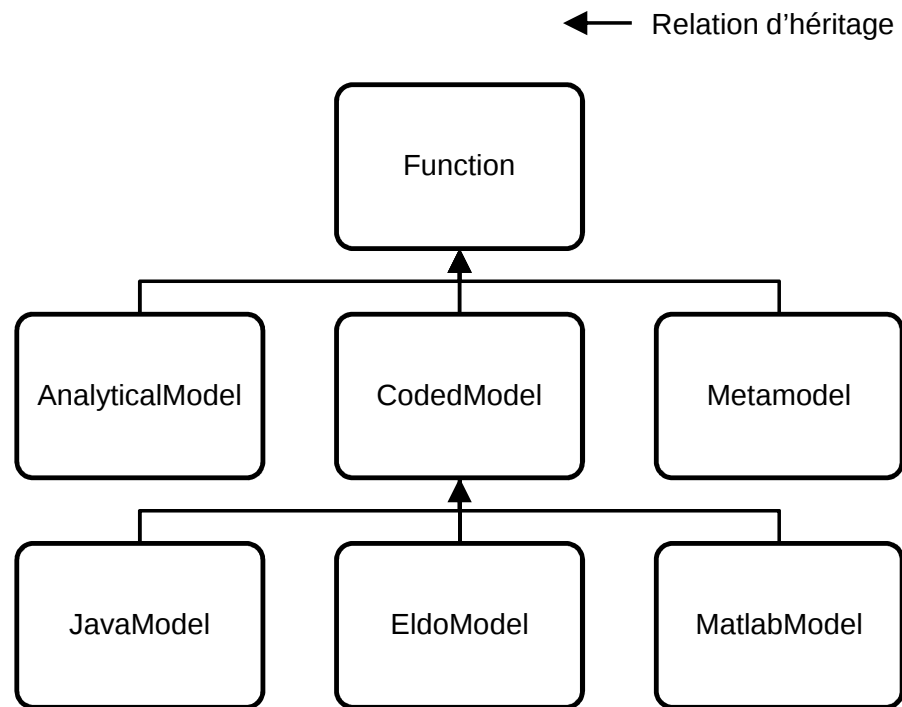


Figure 7.5 : Représentation UML simplifiée de la classe mère *Function* et des classes dérivées

La construction des méta-modèles. La construction des modèles polynomiaux est effectuée grâce à la classe *MetamodelingFactory*. Cette classe met donc en œuvre les plans d'expériences décrits par la classe *DOE* et se charge entre autres de réaliser les simulations électriques (cf. Figure 7.6). Par ailleurs, la classe *MetamodelingFactory* prend en charge la sauvegarde des résultats des simulations en vue d'une réutilisation ultérieure (voir §5.2.5).

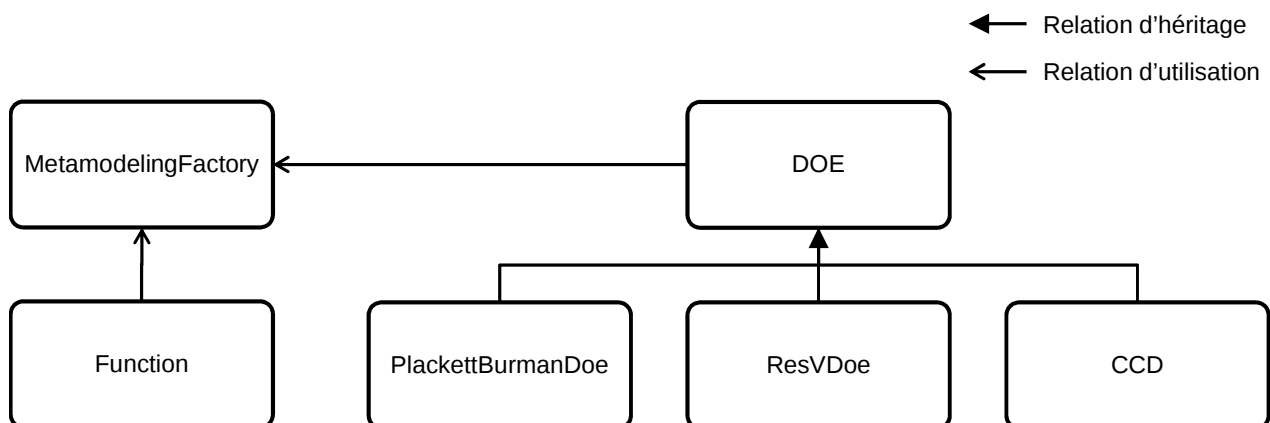


Figure 7.6 : Représentation UML simplifiée de la classe *MetamodelingFactory* et de ses interactions avec les autres classes

Les algorithmes d'optimisation par intervalles sont codés par l'intermédiaire de deux classes : la première décrit l'algorithme de base tandis que la seconde incorpore en plus les contracteurs.

Les plans de conception. Enfin, une classe *DesignPlan* permet l'initialisation et le lancement de l'algorithme d'optimisation par intervalles qui lui-même appelle la classe *MetamodelingFactory* pour construire les méta-modèles. Le flot de la méthode de conception robuste est illustré sur la Figure 7.7.

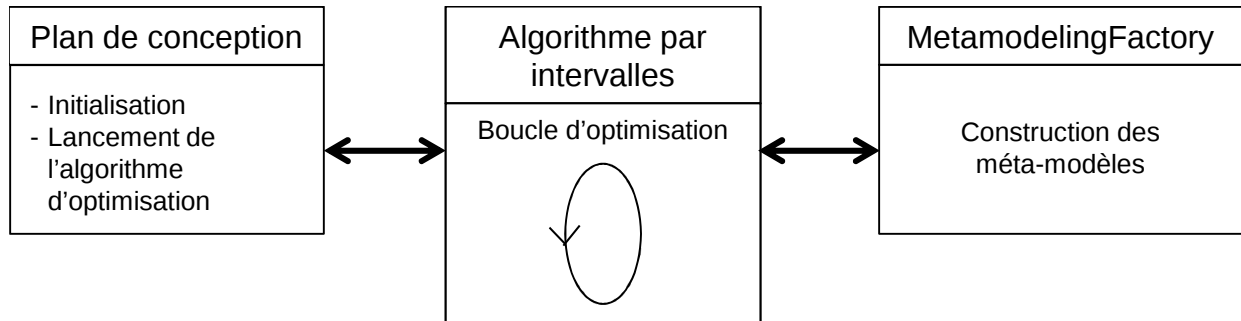


Figure 7.7 : Flot de la conception robuste