



**HAL**  
open science

# OPTIMISATION STATISTIQUE DU RENDEMENT PARAMETRIQUE DE MICROSYSTEMES (MEMS)

Flavien Delauche

► **To cite this version:**

Flavien Delauche. OPTIMISATION STATISTIQUE DU RENDEMENT PARAMETRIQUE DE MICROSYSTEMES (MEMS). Micro et nanotechnologies/Microélectronique. Université Montpellier II - Sciences et Techniques du Languedoc, 2003. Français. NNT : . tel-00005359

**HAL Id: tel-00005359**

**<https://theses.hal.science/tel-00005359>**

Submitted on 16 Mar 2004

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

**UNIVERSITE MONTPELLIER II  
SCIENCES ET TECHNIQUES DU LANGUEDOC**

**T H E S E**

Pour obtenir le grade de

**DOCTEUR DE L'UNIVERSITE MONTPELLIER II**

***Discipline : Micro-électronique***  
***Formation Doctorale : Systèmes Automatiques et Micro-électronique***  
***École doctorale : Information, Structures, Systèmes***

présentée et soutenue publiquement

par

**Flavien DELAUCHE**

le 23 mai 2003

**Titre :**

**OPTIMISATION STATISTIQUE DU  
RENDEMENT PARAMETRIQUE  
DE MICROSYSTEMES (MEMS)**

**JURY**

|   |                      |
|---|----------------------|
| M. Michel ROBERT, Professeur, Université Montpellier II             | , Président          |
| M. Christian DUFAZA, Professeur, Université Aix-Marseille I         | , Directeur de thèse |
| M. Skandar BASROUR, Professeur, Université Joseph Fourier, Grenoble | , Rapporteur         |
| M. Yann HU, Professeur, Université Aix-Marseille III                | , Rapporteur         |
| M. Bachar AFFOUR, Docteur, MEMSCAP Crolles                          | , Examineur          |



## REMERCIEMENTS

Je tiens en premier lieu à remercier Michel Habib, directeur du LIRMM (CNRS), mon laboratoire d'attache, ainsi que Jean-Michel Karam, PDG de MEMSCAP, l'entreprise d'accueil dans laquelle s'est déroulée la plus grande partie de ma thèse CIFRE, pour avoir rendu possible cette thèse, à cheval entre la recherche et l'industrie d'une technologie émergente.

Merci bien sûr à mes tuteurs, le professeur Christian Dufaza (Université Aix-Marseille I) et Bachar Affour (MEMSCAP), qui ont pu rester disponibles malgré les difficultés à encadrer une thèse loin du laboratoire d'attache et au sein d'une start-up évoluant dans un milieu très concurrentiel et instable durant ces 3 années.

Je remercie aussi les autres membres du jury : les professeurs Skandar Basrou et Yann Hu pour avoir pris le temps d'examiner ce rapport, et le professeur Michel Robert pour avoir accepté la présidence de ce jury.

Je remercie enfin chaleureusement l'ensemble des ingénieurs de MEMSCAP avec lesquels j'ai travaillé durant cette thèse, qu'ils soient des groupes CAO, RF ou optique : l'optimisation est par chance un domaine très fédérateur. Je tiens à remercier particulièrement Marie-Pierre Brutails qui m'a aidé à structurer l'implémentation informatique de l'algorithme développé dans le cadre de cette thèse.



# TABLE DES MATIERES

|  |           |
|--|-----------|
| <u>REMERCIEMENTS</u>   | <u>3</u>  |
| <u>TABLE DES MATIERES</u>                                    | <u>5</u>  |
| <u>LISTE DES FIGURES</u>                                     | <u>9</u>  |
| <u>LISTE DES TABLEAUX</u>                                    | <u>13</u> |
| <u>INTRODUCTION GENERALE</u>                                 | <u>17</u> |
| <u>1. MICROSYSTEMES ET RENDEMENT</u>                         | <u>23</u> |
| <u>1.1 Les microsystemes</u>                                 | <u>23</u> |
| 1.1.1 Technologies MEMS                                      | 23        |
| a) Le micro-usinage en volume                                | 23        |
| b) Le micro-usinage en surface                               | 24        |
| c) Sources et conséquences des défauts de fabrication        | 25        |
| 1.1.2 Caractérisation et Fiabilité                           | 26        |
| a) Caractérisation   | 26        |
| b) Analyse de défaillance                                    | 26        |
| 1.1.3 Encapsulation  | 26        |
| 1.1.4 Applications et spécificités technologiques            | 27        |
| a) Fluidique   | 27        |
| b) Mécanique   | 27        |
| c) Optique   | 28        |
| d) Applications Radio-Fréquence (RF)                         | 29        |
| 1.1.5 Logiciels de CAO                                       | 29        |
| a) Outils disponibles  | 29        |
| b) Demande des concepteurs MEMS                              | 30        |
| <u>1.2 Le rendement</u>                                      | <u>30</u> |
| 1.2.1 Notions de rendement                                   | 30        |
| a) Défauts catastrophiques et paramétriques                  | 31        |
| b) Amélioration en amont de la fabrication                   | 31        |
| c) Amélioration en ligne                                     | 32        |
| 1.2.2 Rendement et CAO microélectronique                     | 33        |
| a) Optimisation nominale                                     | 33        |
| b) Optimisation du rendement                                 | 33        |
| 1.2.3 Rendement et industrie                                 | 35        |
| <u>Références</u>  | <u>37</u> |
| <u>2. OPTIMISATION STATISTIQUE DU RENDEMENT PARAMETRIQUE</u> | <u>43</u> |
| <u>2.1 Rendement : définitions</u>                           | <u>43</u> |
| 2.1.1 Les performances ciblées                               | 43        |

|  |                  |
|--|------------------|
| a) Courbes cibles  | 44               |
| b) Limites et gabarits   | 44               |
| c) Fonctions mathématiques                                       | 44               |
| 2.1.2 Le flot d'optimisation                                     | 45               |
| <b><u>2.2</u></b> <b><u>Différentes sortes de paramètres</u></b> | <b><u>45</u></b> |
| 2.2.1 Paramètres de dessin                                       | 45               |
| 2.2.2 Paramètres technologiques                                  | 46               |
| 2.2.3 Paramètres mixtes  | 46               |
| 2.2.4 Paramètres et espaces vectoriels                           | 47               |
| <b><u>2.3</u></b> <b><u>Estimation du rendement</u></b>          | <b><u>49</u></b> |
| 2.3.1 Méthodes déterministes                                     | 49               |
| 2.3.2 Echantillonnage statistique                                | 50               |
| a) Les lois de probabilité                                       | 50               |
| b) Interprétation de l'analyse de Monte-Carlo                    | 51               |
| c) Variantes de l'analyse de Monte-Carlo                         | 53               |
| 2.3.3 Méthodes géométriques                                      | 55               |
| a) Approximation polyédrique ( <i>simplicial approximation</i> ) | 55               |
| b) Approximation ellipsoïdale                                    | 56               |
| c) Exploration orthogonale                                       | 56               |
| d) Exploration radiale   | 56               |
| e) Intégrale de surface  | 57               |
| 2.3.4 Méthode des Pires Cas                                      | 58               |
| <b><u>2.4</u></b> <b><u>Optimisation du rendement</u></b>        | <b><u>59</u></b> |
| 2.4.1 Mesures du rendement et de la qualité                      | 59               |
| a) Indice de capabilité  | 59               |
| b) Fonction Perte de Taguchi                                     | 61               |
| c) Autres fonctions coût   | 63               |
| 2.4.2 Méthodes d'optimisation du rendement                       | 63               |
| a) Algorithmes d'optimisation nominale                           | 64               |
| b) Algorithmes d'optimisation statistique                        | 65               |
| c) Méthodes d'optimisation géométriques                          | 67               |
| 2.4.3 Plans d'expériences  | 69               |
| a) Notion de qualité   | 69               |
| b) Notions sur les plans d'expériences                           | 69               |
| <b><u>Références</u></b>   | <b><u>81</u></b> |
| <b><u>3. NOUVELLE STRATEGIE</u></b>                              | <b><u>87</u></b> |
| <b><u>3.1</u></b> <b><u>Justification</u></b>                    | <b><u>87</u></b> |
| 3.1.1 Optimisation en deux temps                                 | 87               |
| 3.1.2 Nature statistique du problème                             | 88               |
| a) Circuits discrets et circuit intégrés                         | 88               |
| b) Cas des circuits discrets                                     | 89               |
| c) Cas des circuits intégrés                                     | 90               |
| 3.1.3 Utilisation des plans multiples de Taguchi                 | 91               |
| a) Calcul de la "moyenne"  | 92               |

|  |                   |
|--|-------------------|
| b) Calcul de la dispersion   | 93                |
| <b><u>3.2</u></b> <b><u>Spécifications et performances</u></b>                       | <b><u>94</u></b>  |
| <b><u>3.3</u></b> <b><u>Estimation du rendement</u></b>                              | <b><u>94</u></b>  |
| <b><u>3.4</u></b> <b><u>Réduction de la variabilité d'une performance</u></b>        | <b><u>94</u></b>  |
| 3.4.1 Réduction préalable du nombre de facteurs                                      | 95                |
| 3.4.2 Algorithme de réduction de la variabilité                                      | 95                |
| a) Pas d'incrémentation  | 96                |
| b) Diminuer le nombre de facteurs  | 97                |
| c) Facteurs non fixés  | 99                |
| d) Critères d'arrêt  | 99                |
| e) Organigramme  | 99                |
| <b><u>3.5</u></b> <b><u>Ajustement de la réponse moyenne du système</u></b>          | <b><u>100</u></b> |
| 3.5.1 Optimisation multi-objectifs : anticiper les problèmes à venir                 | 101               |
| 3.5.2 Choix de la fonction erreur  | 102               |
| 3.5.3 Algorithme de l'ajustement de la valeur moyenne d'une performance              | 104               |
| a) Pas d'incrémentation  | 104               |
| b) Diminuer le nombre de facteurs  | 104               |
| c) Critères d'arrêt  | 104               |
| 3.5.4 Les différences entre les deux étapes  | 105               |
| 3.5.5 Organigramme de l'ajustement de la valeur moyenne de la réponse                | 105               |
| <b><u>3.6</u></b> <b><u>Choix des plans d'expériences</u></b>                        | <b><u>106</u></b> |
| <b><u>3.7</u></b> <b><u>Optimisation de plusieurs performances simultanément</u></b> | <b><u>107</u></b> |
| <b><u>4. IMPLEMENTATION DE L'ALGORITHME, APPLICATIONS ET RESULTATS</u></b>           | <b><u>111</u></b> |
| <b><u>4.1</u></b> <b><u>Modélisation des MEMS</u></b>                                | <b><u>111</u></b> |
| 4.1.1 Difficultés de modélisation et de conception                                   | 111               |
| a) Simulations multi-domaines  | 111               |
| b) Importance de l'usure   | 112               |
| c) Problèmes pour une technologie donnée   | 112               |
| d) Le device mismatch  | 113               |
| 4.1.2 Modèles Spice  | 114               |
| 4.1.3 Modèles comportementaux  | 114               |
| 4.1.4 Modèles FEM  | 116               |
| 4.1.5 Choix de nos modèles   | 117               |
| <b><u>4.2</u></b> <b><u>Considération de l'informatique</u></b>                      | <b><u>117</u></b> |
| 4.2.1 Environnement  | 117               |
| 4.2.2 Structure implémentée  | 118               |
| a) Choix des classes   | 118               |
| b) Les paramètres mixtes : un point sensible   | 119               |
| c) Gestion des fichiers  | 120               |
| d) Appel de programmes extérieurs  | 120               |
| <b><u>4.3</u></b> <b><u>Interface du logiciel</u></b>                                | <b><u>120</u></b> |

|                                    |  |            |
|------------------------------------|--|------------|
| <u>4.4</u>                         | <u>Exemple : Filtre passe bande RLC</u>              | <u>122</u> |
| 4.4.1                              | Saisie du problème                                   | 122        |
|                                    | a) Topologie du circuit                              | 122        |
|                                    | b) Choix des paramètres et de leurs caractéristiques | 123        |
|                                    | c) Choix des paramètres de l'algorithme              | 124        |
|                                    | d) Choix des spécifications                          | 124        |
| 4.4.2                              | Suivi du processus d'optimisation                    | 125        |
| 4.4.3                              | Résultats  | 127        |
|                                    | a) Interprétation des chiffres                       | 127        |
|                                    | b) Interprétation des courbes                        | 128        |
|                                    | c) Conclusion  | 130        |
| <u>4.5</u>                         | <u>Exemple : Microrésonateur MEMS</u>                | <u>131</u> |
| 4.5.1                              | Saisie du problème                                   | 132        |
|                                    | a) Topologie du MEMS                                 | 133        |
|                                    | b) Choix des paramètres et de leurs caractéristiques | 134        |
|                                    | c) Choix des spécifications                          | 135        |
| 4.5.2                              | Suivi du processus d'optimisation                    | 136        |
| 4.5.3                              | Résultats  | 137        |
| <u>4.6</u>                         | <u>Exemple : Inductance MEMS</u>                     | <u>138</u> |
| <u>4.7</u>                         | <u>Variante : Optimiseur nominal</u>                 | <u>141</u> |
|                                    | <u>Références</u>                                    | <u>143</u> |
| <u>CONCLUSIONS ET PERSPECTIVES</u> |  | <u>148</u> |
| <u>ANNEXES</u>                     |  | <u>153</u> |
| <u>I.</u>                          | <u>Classes utilisées</u>                             | <u>153</u> |
| <u>II.</u>                         | <u>Fichiers utilisés</u>                             | <u>155</u> |
| <u>III.</u>                        | <u>Les filtres à peignes électrostatiques</u>        | <u>157</u> |
| <u>IV.</u>                         | <u>Ressources</u>                                    | <u>159</u> |
| <u>V.</u>                          | <u>Publications</u>                                  | <u>160</u> |

## LISTE DES FIGURES

|   |    |
|---|----|
| Figure 1-1 : Micro-usinage en volume  | 23 |
| Figure 1-2 : Capteur magnétique réalisé par micro-usinage en volume [LATO99]  | 24 |
| Figure 1-3 : Micro-usinage en surface   | 24 |
| Figure 1-4 : Technologie LIGA   | 25 |
| Figure 1-5 : Principe de la micropompe  | 27 |
| Figure 1-6 : Accéléromètre réalisé en usinage de surface  | 28 |
| Figure 1-7 : La technologie DLP de Texas Instrument   | 28 |
| Figure 1-8 : Des masques au modèle comportemental   | 30 |
| Figure 1-9 : Phénomène de stiction entre un engrenage et une crémaillère  | 31 |
| Figure 1-10 : L'optimisation nominale : ajuster les performances d'un système   | 33 |
| Figure 1-11 : Amélioration du rendement et de la robustesse   | 35 |
| Figure 1-12 : Faible dispersion d'un procédé de fabrication   | 36 |
| Figure 2-1 : Analyse AC d'un MEMS et sa courbe cible associée   | 44 |
| Figure 2-2 : Analyse AC d'un MEMS et limites associées  | 44 |
| Figure 2-3 : Flot d'optimisation des performances d'un MEMS   | 45 |
| Figure 2-4 : Structure à peignes électrostatiques   | 46 |
| Figure 2-5 : La largeur d'une piste est un paramètre mixte  | 46 |
| Figure 2-6 : Représentation des variables, données de l'optimisation  | 47 |
| Figure 2-7 : Région de Tolérance (Rt)   | 48 |
| Figure 2-8 : Région d'Acceptabilité (Ra)  | 48 |
| Figure 2-9 : Recouvrement de Ra et Rt pour un MEMS dans les spécifications  | 48 |
| Figure 2-10 : Les différents types de méthodes d'estimation du rendement  | 49 |
| Figure 2-11 : L'intégrale d'une gaussienne sur l'intervalle $[-\infty, +\infty]$ représente 95,4% de l'intégrale sur $[-\infty, +\infty]$ | 51 |
| Figure 2-12 : Abaque du rendement réel en fonction du rendement estimé et du nombre de simulations considéré                              | 53 |
| Figure 2-13 : Echantillonnage stratifié et méthode de régionalisation   | 54 |
| Figure 2-14 : Méthodes d'estimation du rendement basées sur la recherche de la région d'acceptabilité Ra                                  | 55 |
| Figure 2-15 : Recherche de Ra par approximation polyédrique   | 55 |
| Figure 2-16 : Approximation de Ra par un ellipsoïde et par des fragments d'ellipsoïdes  | 56 |
| Figure 2-17 : Estimation du rendement par exploration orthogonale   | 56 |

|  |     |
|--|-----|
| Figure 2-18 : Estimation du rendement par exploration radiale  | 57  |
| Figure 2-19 : Estimation du rendement par la méthode de l'intégrale de surface                         | 57  |
| Figure 2-20 : Sommet correspondant au pire cas dans un espace à trois dimensions                       | 58  |
| Figure 2-21 : Indice de capabilité d'une performance   | 59  |
| Figure 2-22 : Indice de capabilité $C_p = 1$ et $C_p = 2$  | 60  |
| Figure 2-23 : Deux fonctions Perte de même valeur moyenne dans le cas d'un échantillon                 | 62  |
| Figure 2-24 : Fonction Perte dans le cas où l'optimum est un minimum ou un maximum                     | 62  |
| Figure 2-25 : Méthodes d'optimisation du rendement   | 64  |
| Figure 2-26 : Optimisation du rendement par l'algorithme du centre de gravité                          | 65  |
| Figure 2-27 : Optimisation du rendement par l'échantillonnage paramétrique                             | 66  |
| Figure 2-28 : Réalisation d'une coupe pour un point violant les spécifications                         | 67  |
| Figure 2-29 : Centrage d'un circuit par la méthode de l'exploration radiale                            | 68  |
| Figure 2-30 : Hypersphère inscrite dans le polyèdre définissant Ra                                     | 68  |
| Figure 2-31 : Utilisation de facteurs centrés pour construire un polynôme                              | 73  |
| Figure 2-32 : Plans d'expériences usuels   | 74  |
| Figure 2-33 : Corrélation entre facteurs de bruit  | 77  |
| Figure 3-1 : Organigramme de l'algorithme d'optimisation du rendement                                  | 87  |
| Figure 3-2 : Une technologie jeune est pénalisante pour obtenir une faible dispersion des performances | 88  |
| Figure 3-3 : Effet des termes quadratiques sur la réduction de la variance                             | 90  |
| Figure 3-4 : Simulations AC d'un passe-bas en présence de bruit  | 92  |
| Figure 3-5 : Calcul de l'erreur sur une simulation   | 92  |
| Figure 3-6 : Déplacer la région de tolérance pour minimiser la dispersion                              | 96  |
| Figure 3-7 : Pas d'incrémentations   | 96  |
| Figure 3-8 : Détermination d'un pas minimum  | 97  |
| Figure 3-9 : Oscillations autour d'un minimum  | 98  |
| Figure 3-10 : Valeur limite atteinte pour un facteur de dessin   | 99  |
| Figure 3-11 : Organigramme de la réduction de la variabilité   | 100 |
| Figure 3-12 : L'erreur est proportionnelle à l'amplitude de la réponse                                 | 102 |
| Figure 3-13 : Combinaison de contraintes pour une optimisation des performances                        | 102 |
| Figure 3-14 : Organigramme de l'ajustement de la valeur moyenne d'une performance                      | 106 |
| Figure 4-1 : La forme d'une simple poutre est fonction de la technologie utilisée                      | 112 |
| Figure 4-2 : La partie non protégée de la poutre de droite subit une sur-gravure                       | 113 |
| Figure 4-3 : Dispersion des paramètres technologiques  | 113 |
| Figure 4-4 : Choix de la valeur d'un facteur de bruit  | 114 |
| Figure 4-5 : Une plaque : composant élémentaire orienté  | 115 |

|  |     |
|--|-----|
| Figure 4-6 : Netlist et modèle comportant des paramètres de bruit <i>intégrés</i> et <i>discrets</i>         | 116 |
| Figure 4-7 : La réécriture de la netlist doit être simple et rapide  | 118 |
| Figure 4-8 : Diagramme de classes des données de base  | 119 |
| Figure 4-9 : Les classes sont rangées par fonction   | 119 |
| Figure 4-10 : Interface de sélection des paramètres manipulés lors de l'optimisation                         | 121 |
| Figure 4-11 : Circuit RLC à optimiser  | 123 |
| Figure 4-12 : Fichier de suivi de l'évolution des valeurs d'un facteur de dessin                             | 126 |
| Figure 4-13 : Tracé des spécifications et de l'analyse de Monte-Carlo pour l'estimation du rendement initial | 128 |
| Figure 4-14 : Courbes des analyses de Monte-Carlo lors du processus d'optimisation                           | 129 |
| Figure 4-15 : Visualisation de l'évolution de chaque paire (   | 130 |
| Figure 4-16 : Nécessité d'une étape <i>post-process</i> pour libérer les structures                          | 131 |
| Figure 4-17 : Filtre MEMS réalisé avec une technologie AMS, testé au CIME                                    | 132 |
| Figure 4-18 : Elaboration d'une structure MEMS avec MemsMaster   | 133 |
| Figure 4-19 : Paramètres de dessin et de bruit du MEMS   | 135 |
| Figure 4-20 : Optimisation du rendement avec une analyse temporelle  | 137 |
| Figure 4-21 : Paramètres géométriques d'une inductance MEMS  | 138 |
| Figure 4-22 : Principaux phénomènes physiques liés à une inductance  | 138 |
| Figure 4-23 : Description d'une technologie MEMS pour une inductance   | 140 |
| Figure 0-1 : Résonnant Gate Transistor   | 157 |
| Figure 0-2 : Principe des structures de Nguyen résonnant verticalement                                       | 157 |
| Figure 0-3 : Structure résonante à peignes électrostatique latéraux [NGUY98]                                 | 158 |
| Figure 0-4 : Variantes de filtres MEMS   | 158 |



## LISTE DES TABLEAUX

|   |     |
|---|-----|
| Tableau 2-1 : Valeur de l'intégrale en fonction de l'intervalle considéré     | 51  |
| Tableau 2-2 : Erreur sur le rendement estimé                                  | 53  |
| Tableau 2-3 : Valeurs conseillées de l'indice de capabilité $C_p$             | 60  |
| Tableau 2-4 : Valeurs conseillées de l'indice de capabilité $C_{pk}$          | 61  |
| Tableau 2-5 : Principaux cas et rapports signal/bruit associés (Taguchi)      | 63  |
| Tableau 2-6 : Valeurs plausibles pour les facteurs : niveaux                  | 71  |
| Tableau 2-7 : Matrice d'un plan d'expérience pour 2 facteurs à 2 niveaux      | 71  |
| Tableau 2-8 : Plan produit de Taguchi   | 79  |
| Tableau 3-1 : Plan produit : cas où la cible est un vecteur                   | 93  |
| Tableau 3-2 : Choix des plans d'expériences en fonction du nombre de facteurs | 107 |



# INTRODUCTION GENERALE



## INTRODUCTION GENERALE

Le travail effectué durant cette thèse et exposé dans ce document porte sur l'amélioration du rendement de fabrication de microsystèmes, que l'on notera par la suite MEMS (Micro Electro Mechanical Systems). Il a consisté en la mise au point d'un algorithme d'optimisation du rendement des MEMS puis son implémentation et son évaluation. Ce travail a été mené en collaboration avec le laboratoire d'Informatique de Robotique et de Microélectronique de Montpellier (LIRMM) et la société MEMSCAP, basée à Bernin, dans la région grenobloise.

Les microtechnologies sont un ensemble de techniques de fabrication permettant de réaliser des structures de l'ordre du micromètre. C'est typiquement le cas des MEMS qui sont touchés par l'engouement des industriels pour la miniaturisation. Les MEMS ont la particularité de pouvoir intégrer sur un même substrat des parties aussi différentes que des capteurs, des actionneurs, ou une partie microélectronique de commande et de traitement de l'information. C'est la cohabitation sur une surface très réduite de ces différentes technologies dédiées à divers domaines de la physique (électronique, mécanique, optique, électro et magnétostatique, chimie...) qui rend les MEMS si attractifs et si complexes à réaliser. Le principal avantage est la réduction de l'encombrement et de la masse du dispositif, ce qui est très important pour de nombreuses applications médicales ou spatiales. L'utilisation de technologies proches de la microélectronique permet de plus d'envisager une production de masse à moindre frais. Concernant les capteurs, la réduction des parasites habituellement dus à la connectique est très importante. En ce qui concerne les actionneurs, le fait que les puces puissent agir physiquement sur leur environnement avec une grande précision annonce l'ère des micromachines.

La première publication concernant un MEMS opérationnel date de 1967 [NATH67] (page 40), 10 ans après l'invention du circuit intégré à base de transistors, et fait globalement appel aux mêmes techniques de fabrication. Cependant, l'intérêt pour les MEMS sera plus long à venir et concernera initialement les capteurs au début des années 80. Les produits MEMS les plus populaires sont aujourd'hui les têtes d'imprimante à jet d'encre [HP-net] (page 38) et les accéléromètres [AD-net] (page 37) qui commandent l'ouverture des airbags des voitures. Des applications plus complexes comme les projecteurs numériques [TI-net] (page 40), où les bobines sont remplacées par des supports

numériques, prennent le relais avec des retombées non négligeables. On le voit bien, les MEMS sont maintenant assez matures pour les applications industrielles.

Plus que jamais, la concurrence entre les entreprises se joue au niveau international. Les cycles de vie des produits se raccourcissent de manière parfois spectaculaire, notamment pour les produits intégrant des technologies de pointe. Pour que ces derniers soient rentabilisés avant d'être dépassés, il convient de réduire au maximum le temps écoulé entre leur conception et leur mise sur le marché. C'est pourquoi il est primordial de développer un environnement de CAO (Conception Assistée par Ordinateur) dédié aux MEMS. Les problèmes de modélisation et de simulation lors d'interactions entre divers domaines de la physique différencient notamment les MEMS de la microélectronique classique. De plus, la jeunesse et la complexité des technologies MEMS rendent leur utilisation délicate et les rendements incertains.

C'est sur cette thématique du rendement que se penche ce manuscrit, et plus particulièrement sur son amélioration en phase de conception des structures MEMS. Il s'agit de fournir au concepteur un outil permettant d'optimiser les paramètres géométriques des MEMS en vue de minimiser l'impact des légères variations aléatoires que pourrait avoir la technologie employée sur les performances du MEMS [SANC00] (page 40). Cet outil constitue donc un pont entre le monde des concepteurs et celui des technologues. Il s'agit ici de gagner les quelques pourcents qui feront la différence avec un concurrent. A titre de comparaison, Cadence, fournisseur d'outils de CAO pour la microélectronique, ne propose un tel outil que depuis juillet 2002. Même si le concept est bien plus ancien, il n'a jamais percé en microélectronique auparavant.

MEMSCAP a été fondée en 1997 et dispose depuis 2002 de ses propres locaux et d'une fonderie : c'est un environnement bien adapté pour un tel travail, les technologues, les concepteurs de MEMS et les programmeurs d'outils de CAO étant tous disponibles au sein de l'entreprise. Le faible succès des logiciels d'optimisation statistique pour les microtechnologies est dû à deux causes principales. Tout d'abord, la caractérisation statistique de la technologie employée est primordiale, mais ces données étant confidentielles, il faut un haut niveau de confiance entre le fondeur et le concepteur du circuit pour que ce dernier puisse s'en servir. Ensuite, les concepteurs ont une certaine réticence à s'immiscer dans le monde des statistiques; la facilité d'utilisation sera donc l'un des critères clés de l'évaluation du programme.

Le LIRMM s'intéresse aux MEMS depuis 1995, notamment à leur conception, à leur test, à la caractérisation de processus technologiques, ainsi donc qu'à la CAO. La

microélectronique et les MEMS sont intimement liés, il est donc normal pour un laboratoire tel que le LIRMM de s'impliquer dans ce domaine en pleine expansion.

Le travail s'est déroulé en quatre étapes :

- Etat de l'art des technologies MEMS et des bases du rendement en microtechnologies.
- L'optimisation statistique : énoncé du problème, aperçu des méthodes employées et proposition d'une nouvelle approche.
- Implémentation de cet algorithme en langage Java.
- Evaluation de l'algorithme sur des structures tests et analyse des résultats.

Le premier chapitre reprend les techniques de fabrication des MEMS ainsi que les principaux types d'applications. Puis nous traitons des différentes approches employées pour améliorer le rendement. Un état de l'art de la CAO pour les MEMS et quelques notions sur le rendement en microélectronique permettent de bien positionner le problème.

Le second chapitre donne une définition plus précise de ce qu'est le rendement dans le cadre de notre travail et apporte des précisions sur la nature des paramètres manipulés lors de son étude. Nous traitons ensuite le problème de l'estimation du rendement et passons en revue les principales méthodes d'optimisation statistique. Nous rappelons aussi les bases mathématiques des *plans d'expériences* que l'on retrouve très souvent dans des contextes d'optimisation du rendement.

Le troisième chapitre est consacré à la mise au point d'une approche originale permettant l'optimisation statistique du rendement paramétrique des MEMS. Utilisant des plans d'expériences, l'algorithme que nous présentons agit en deux temps : il réduit d'abord la variabilité des performances, puis minimise les différences entre les performances et les spécifications données.

Le quatrième chapitre de ce manuscrit relate l'implémentation en langage Java de cet algorithme et son utilisation. Nous abordons aussi les problèmes de simulation et de modélisation des MEMS et justifions les choix réalisés lors de l'élaboration du programme. Deux exemples sont développés pour expérimenter l'optimisation : un circuit discret RLC et un MEMS, pour lesquels les résultats obtenus sont analysés.

Finalement, la conclusion générale fait la synthèse de ces trois années de thèse, et revient sur les résultats obtenus.



# Chapitre 1

## MICROSYSTEMES ET RENDEMENT



## 1. MICROSYSTEMES ET RENDEMENT

Dans ce chapitre, nous donnons les bases des technologies MEMS. Nous pouvons ainsi avoir un aperçu de leur complexité ainsi que de leurs imperfections qui sont à la source de pertes de rendement ultérieures. Ce rendement sera lui aussi défini, et nous présenterons les outils de CAO qui lui sont dédiés.

### 1.1 LES MICROSYSTEMES

Nous donnons un aperçu des technologies MEMS les plus classiques, ainsi que de leur principaux défauts, à la base du rendement médiocre des prototypes MEMS. Puis nous verrons que la fiabilité est pour les MEMS un domaine où il y a énormément à faire, de même que pour l'encapsulation. Enfin, nous donnerons en exemple quelques applications qui ont fait leurs preuves.

#### 1.1.1 TECHNOLOGIES MEMS

Il existe deux familles de technologies MEMS qui se démarquent : le micro-usinage en volume et en surface.

##### a) Le micro-usinage en volume

Le *bulk micromachining* permet d'obtenir les formes recherchées pour un MEMS en travaillant le substrat brut [MADO02]. Cela peut se faire de différentes manières (gravures sèches ou humides, etc.). Dans le cas d'une gravure anisotropique, fréquemment utilisée, la vitesse de gravure varie suivant l'orientation des plans attaqués et la sélectivité de la solution de gravure (KOH, TMAH, EDP...). Ces caractéristiques donnent sa forme finale à la topologie du substrat [Figure 1-1].

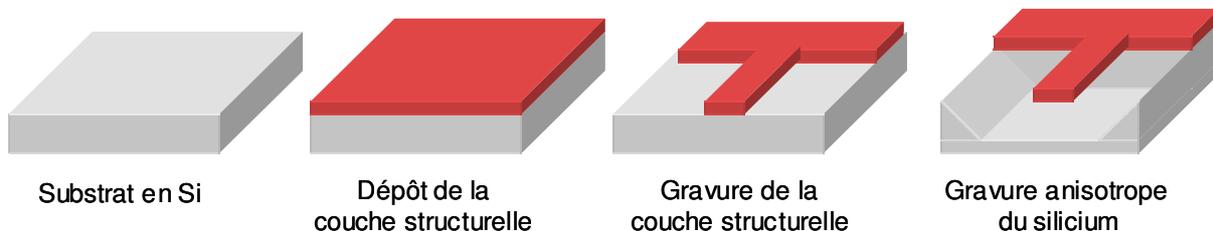
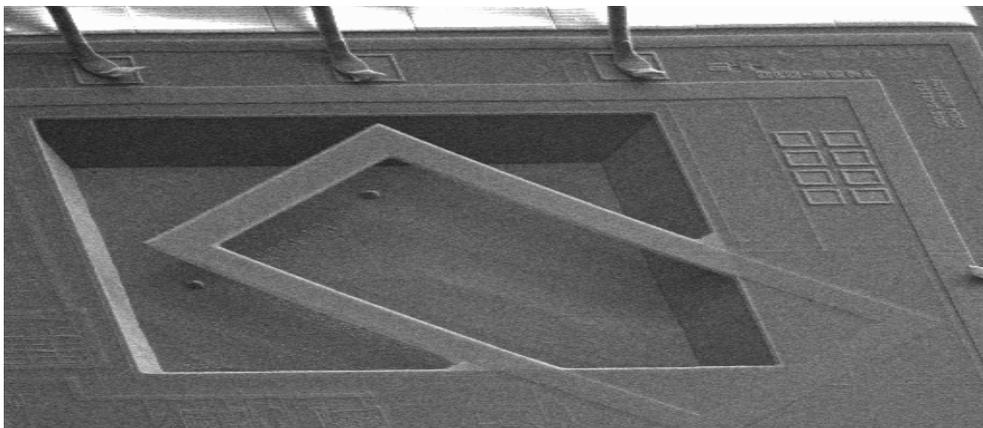


Figure 1-1 : Micro-usinage en volume

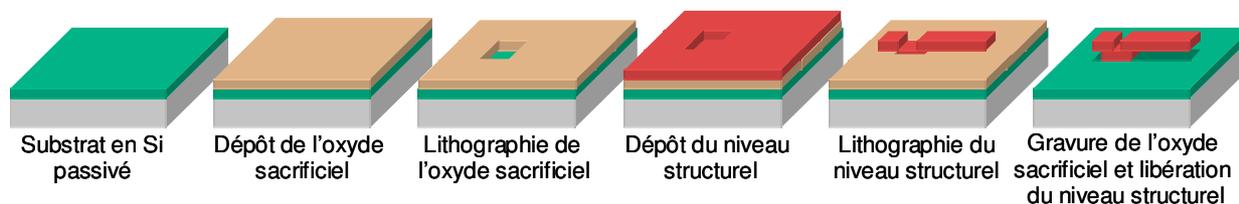
La gravure sèche est compatible avec les technologies que l'on trouve en microélectronique. Il est alors possible que les parties suspendues renferment des lignes de différents conducteurs [Figure 1-2].



**Figure 1-2 : Capteur magnétique réalisé par micro-usinage en volume [LATO99]**

### **b) Le micro-usinage en surface**

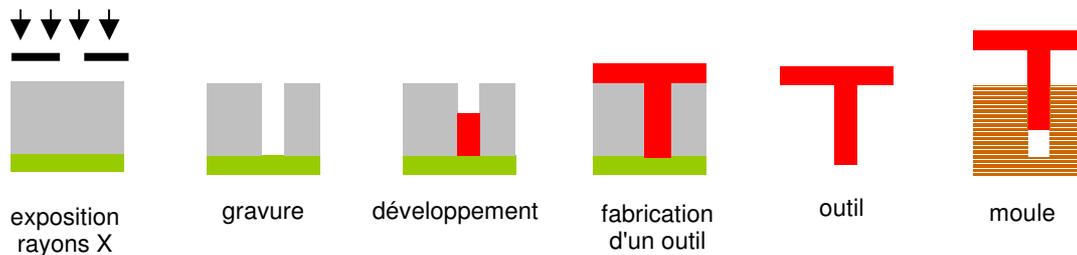
Le *surface micromachining* repose sur la présence de deux types de matériaux au dessus du substrat : structurels et sacrificiels. Ces derniers sont dissous à la fin du procédé de fabrication, libérant les couches structurales. Les couches sacrificielles sont souvent des oxydes ou des résines, et les couches structurales du polysilicium, des métaux ou des isolants non oxydés, tels que le nitrure de silicium [Figure 1-3].



**Figure 1-3 : Micro-usinage en surface**

Certaines de ces technologies sont aujourd'hui assez connues : MUMPs [MUMP] de Cronos (aujourd'hui Memscap), SUMMIT de Sandia [SAND-net], iMEMS d'Analog Devices [AD-net], ou encore celle de Texas Instrument pour réaliser des micromiroirs [TI-net].

La technologie LIGA [Figure 1-4] est considérée à part; elle permet de réaliser des structures de plusieurs centaines de micromètres d'épaisseur, ainsi que des micro-outils ou des moules en métal. Ces derniers permettent de dupliquer une forme à l'envie sur des matériaux plastiques.



**Figure 1-4 : Technologie LIGA**

On a parfois recours à une étape supplémentaire en fin de fabrication : le *wafer bonding*, qui permet de lier deux systèmes conçus sur deux wafers différents en les positionnant face à face et en les liant.

### c) Sources et conséquences des défauts de fabrication

#### c.1 défauts catastrophiques

L'un des principaux défauts de fabrication des technologies MEMS est le collage (*stiction*). Il apparaît lors des dernières étapes de l'usinage en surface lorsque les couches sacrificielles sont dissoutes, ou encore quand le MEMS opère dans un environnement humide. Les couches structurales ont alors tendance à se coller au substrat ou aux éléments voisins. Les raisons de ce comportement sont diverses : la force de capillarité du liquide de rinçage, les forces électrostatiques qui s'appliquent en surface des éléments, la force de Van der Waals... Nous verrons par la suite que ce type de défauts catastrophiques (plus de mouvement possible) réduit le rendement dit fonctionnel. Nous ne chercherons pas à le minimiser dans notre approche CAO. Certaines précautions peuvent réduire ces problèmes : minimiser la force de capillarité du liquide de rinçage, réduire les surfaces de contact entre les structures et le substrat, espacer les structures entre elles quand cela est possible...

#### c.2 défauts paramétriques

Ces défauts de fabrication n'empêchent pas le MEMS de fonctionner mais dégradent ses performances et le rendement dit paramétrique. Ils sont entre autre à l'origine de l'aspect statistique des dispersions des paramètres technologiques. Par exemple, la difficulté de contrôler les gravures ou la dispersion dans les résines photosensibles. Cela entraîne une incertitude quant à la valeur réelle des dimensions géométriques des structures, et donc de leur comportement mécanique. De même, si les étapes de fabrication sont aisément reproductibles, le module d'Young d'un matériau ne sera jamais vraiment deux fois le même, il existe donc aussi une dispersion au niveau des caractéristiques des matériaux. Leur

connaissance permet d'envisager une conception qui tendrait à minimiser leur impact sur les performances du MEMS; c'est l'objet des travaux de cette thèse.

### 1.1.2 CARACTERISATION ET FIABILITE

La caractérisation des matériaux et composants permet de mettre en relief les paramètres critiques pour les performances et fournit les données statistiques associées aux paramètres technologiques qui permettront un meilleur dimensionnement des structures MEMS. L'analyse de défaillance, quant à elle, permet de valider le concept des structures mécaniques en mettant en évidence les mécanismes de défaillance. Un travail est ensuite envisageable sur la géométrie des structures et/ou sur les matériaux [DEAL97].

#### a) Caractérisation

Pour permettre un bon comportement des MEMS sur le long terme, une bonne connaissance des caractéristiques mécaniques des couches minces est nécessaire. On utilise donc des structures de test pour définir leurs propriétés : les *parametric monitors* (PM). Les PM sont des structures généralement très simples (poutres, ressorts) qui permettent une caractérisation des propriétés électriques ou mécaniques des matériaux.

#### b) Analyse de défaillance

Les structures de test pour mettre en évidence les mécanismes de défaillance portent d'autres noms : TCV (Technology Characterization Vehicle) ou SED (Standard Evaluation Devices). Les premières sont généralement des composants de librairie qui servent à mettre en évidence un mécanisme de défaillance particulier attribué à une étape précise du procédé de fabrication. Ils permettent notamment d'estimer l'espérance de vie des composants. Un SED est une version moins complexe de ces composants qui est utilisée pour réaliser des mesures de caractéristiques sur différents wafers et lots [MUKH99] [GUPT97] [CHAN99].

En comparaison avec la microélectronique, les mécanismes de défaillance ne sont ni bien compris, ni faciles à contrôler dans le cadre d'un vieillissement accéléré [MCCL99]. Les principaux sont les suivants : les fractures mécaniques [JONE99], les collages [MAST99], l'usure, la délamination, les problèmes dus à l'environnement (vibrations, chocs, humidité, radiations, particules, température, décharges électrostatiques...), le stress (thermique et résiduel), les capacités parasites, les frottements sur les parties mobiles (*damping*).

### 1.1.3 ENCAPSULATION

L'encapsulation (*packaging*) est aussi importante pour les MEMS que pour les circuits intégrés. Il s'agit d'établir les interconnexions entre le MEMS et d'autres circuits, ainsi que de

maintenir un environnement de fonctionnement adéquat. Aux contraintes existantes déjà pour les circuits intégrés électroniques, il faut rajouter les contraintes propres à des parties mécaniques. Par exemple, on fera encore plus attention aux moyens d'évacuer la chaleur dégagée par la partie électrique et les frottements mécaniques. Le boîtier sert à la fois de support mécanique, de protection contre l'environnement, et de support de connexions avec le monde extérieur. Sa mise au point est cruciale pour un MEMS.

#### 1.1.4 APPLICATIONS ET SPECIFICITES TECHNOLOGIQUES

Il existe énormément de prototypes de MEMS, MOEMS (Micro Opto Electro Mechanical System), BIOMEMS...; peu ont cependant franchi la phase d'industrialisation. Nous donnons en exemple quelques applications MEMS qui démontrent pourtant le fort potentiel des microtechnologies. C'est à la vue de telles données que l'intérêt pour le rendement des MEMS prend sa juste valeur, alors même que l'on considère toujours qu'il s'agit d'une technologie émergente. Parallèlement, le nombre de conférences et de journaux dédiés aux MEMS croît et leur nombre montre l'intérêt que les laboratoires leur accordent désormais. Les principales ressources dans ce domaine sont citées en annexe [p. 159].

##### a) Fluidique

Les têtes à jet d'encre fonctionnent avec des technologies MEMS depuis les années 80 [PELE98]. L'optimisation des micropompes [Figure 1-5] permet de réaliser des caractères toujours plus fins et nets. Mais ces pompes trouvent aussi une application dans le médical, où elles peuvent par exemple être utilisées pour l'injection sous-cutanée d'insuline à faible dose [GADE01]. La technique d'usinage en volume est ici souvent utilisée associée au wafer bonding.

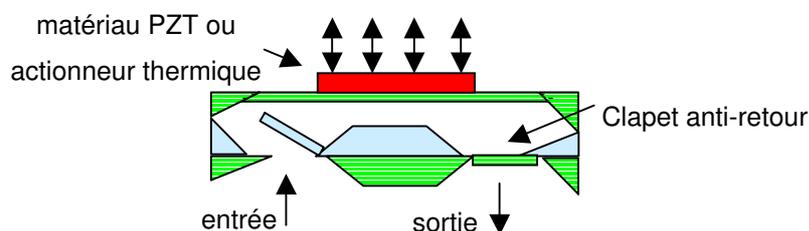
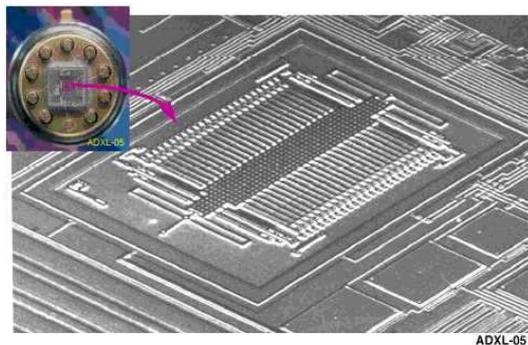


Figure 1-5 : Principe de la micropompe

##### b) Mécanique

Les capteurs de pression existent sur le marché depuis longtemps. Une différence de pression entraîne la déformation d'une membrane qui peut être mesurée. On en déduit la pression extérieure au MEMS. Les accéléromètres [ZHAN98] sont une application plus

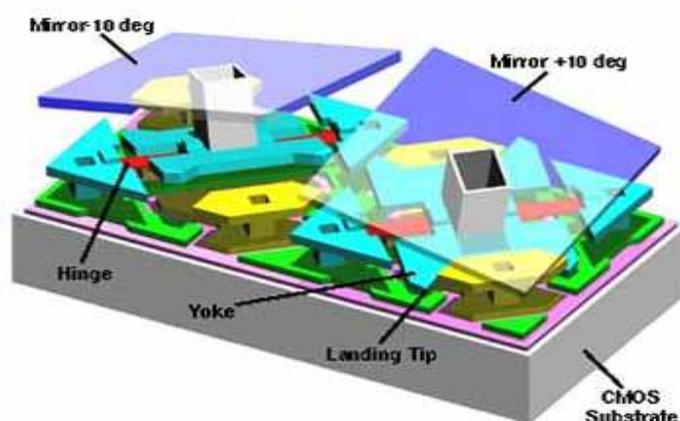
récente dont le succès est aussi très important, notamment dans le domaine de l'automobile, où ils sont utilisés pour commander l'ouverture des airbags en cas de choc et supplantent tous les autres systèmes. Analog Devices utilise l'usinage en surface pour son ADXL [Figure 1-6] [AD-net].



**Figure 1-6 : Accéléromètre réalisé en usinage de surface**

### c) Optique

L'application mise en avant pour les MOEMS est la DLP de Texas Instruments (*Digital Light Processing*) [TI-net]. Son ambition est tout simplement de remplacer les écrans conventionnels de télévision ou d'ordinateur par un ensemble de micromiroirs, associés chacun à un pixel [Figure 1-7]. Leur intérêt est le faible encombrement du moniteur et une résolution de qualité numérique. Ce système est déjà utilisé dans des salles de cinéma par ce que l'on appelle des projecteurs numériques. Les films ne sont plus sur bandes, mais sur des supports d'informations numériques. Ce nouveau marché est, comme celui de l'automobile, des plus prometteur.



**Figure 1-7 : La technologie DLP de Texas Instrument**

### **d) Applications Radio-Fréquence (RF)**

Le marché des télécommunications motive les entreprises impliquées dans les MEMS car il est souvent rentable à court terme, mais il ne connaît pas encore de succès industriel impliquant un MEMS. Les développements dans ce domaine concernent notamment les capacités, les inductances, les transformateurs, les filtres [HECT99] [NGUY98] et les microrelais [LAFO01] [KRUG99]. Ces systèmes ne comportent pas forcément de parties mécaniques, mais le terme de microsystèmes s'élargit de plus en plus et englobe aujourd'hui l'ensemble des systèmes réalisés à partir de microtechnologies.

#### 1.1.5 LOGICIELS DE CAO

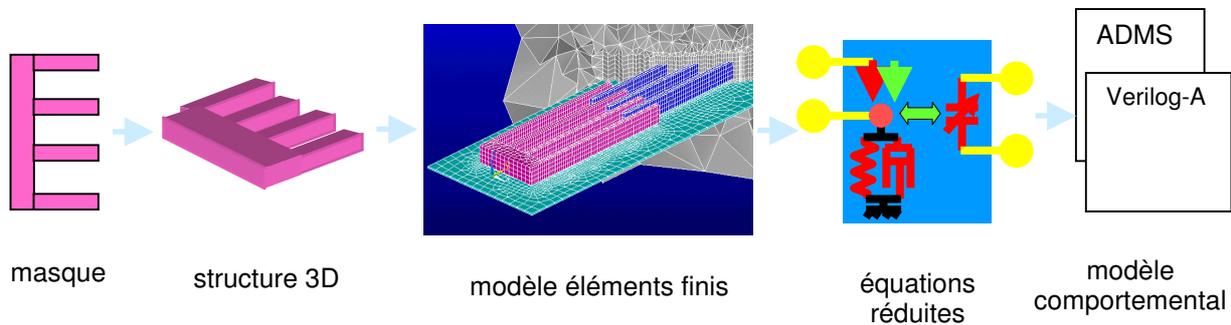
Les MEMS appartiennent à différents domaines : microélectronique, mécanique, électrostatique... Les outils de CAO utilisés lors de leur conception sont donc assez variés, mais pas toujours bien adaptés.

##### **a) Outils disponibles**

Les entreprises proposant des outils de CAO dédiés aux MEMS sont peu nombreuses: Memscap [MEMS-net] et Intellisense [INTE-net] qui disposent aussi d'une fonderie, Coventor [COVE-net] et CFDRC [CFDR-net]. Aucun outil d'optimisation statistique du rendement paramétrique ne figure dans leurs offres. Concernant la partie microélectronique, elle est généralement analogique et conçue dans les environnements des sociétés Cadence ou Mentor Graphics. Concernant la partie purement mécanique, Ansys [ANSY-net] et Abaqus [ABAQ-net] sont utilisés lorsque l'on veut avoir recours aux éléments finis. Pour des applications RF, Agilent [AGIL-net] est souvent l'environnement de conception et Ansoft [ANSO-net] fournit des outils (HFSS) pour les simulations électromagnétiques. Les concepteurs travaillant avec les MOEMS (Micro Opto Electro Mechanical Systems) utilisent habituellement CodeV [CODE-net] pour de gros systèmes (résolution de l'ordre d'une quinzaine de micromètres) assez simples en optique libre (faisceaux parallèles à un axe et sans guides d'ondes). Pour des MOEMS, des logiciels utilisant les méthodes BPM (Beam Propagation Method, intéressant de 10 à 100 intéressant de 1 à 10 nécessaires, mais rendent les temps de simulation extrêmement longs. Rsofts [RSOF-net], Optiwave [OPTI-net] et Alcatel Optronics [ALCA-net] fournissent de tels logiciels. Les solvers multi-physiques (co-simulation) sont donc critiques pour espérer un jour traiter ces différents aspects simultanément, mais de gros investissements sont nécessaires pour les développer, ils sont donc peu nombreux et très lourds à utiliser [WACH99] [NAGL98].

**b) Demande des concepteurs MEMS**

La plupart des outils nécessaires aux concepteurs MEMS existent, mais pas sous le même environnement. Il y a donc une forte demande pour faciliter les transferts et traitements de fichiers par le biais de passerelles entre ces outils. A la base de ces derniers, on trouve les simulateurs de technologie (gravures, etc.) : les technologies MEMS étant encore principalement du ressort de la R&D, la topologie de certaines structures peut être complexe à prévoir, d'où la nécessité de telles simulations permettant de prendre en compte différents types de gravures et de dépôts. Il y a aussi bien sûr les bibliothèques de composants de base ou plus complexes, ainsi que des outils permettant, à partir de la description (masques) d'un MEMS et de sa technologie, de générer la forme correspondante en 3D, afin de réaliser une analyse en éléments finis. Suite à cette analyse, un outil de réduction pourra en tirer un modèle comportemental, facile à insérer dans une netlist et rapide à simuler [Figure 1-8] [MOUL02] [LORE99]. Le concepteur a ainsi la possibilité de créer sa propre bibliothèque de composants avec les modèles comportementaux associés, et non de se contenter des éléments rigides d'une bibliothèque restreinte. Et enfin, on retrouve une demande pour des outils d'optimisation nominale et d'optimisation du rendement, très importants du fait de la complexité des technologies MEMS et donc de leurs dispersions.



**Figure 1-8 : Des masques au modèle comportemental**

**1.2 LE RENDEMENT**

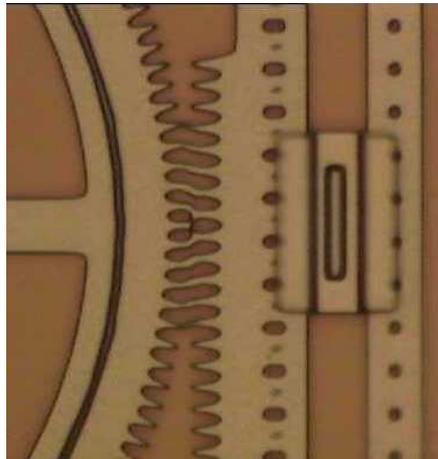
**1.2.1 NOTIONS DE RENDEMENT**

Le rendement peut être évalué à différents stades de la fabrication d'un produit. Il peut être estimé au niveau du projet : les résultats sont-ils à la hauteur de l'investissement ? Au niveau de la fabrication, ce sera le nombre d'éléments fonctionnant parmi ceux fabriqués (rendement de fabrication). On peut donc définir un nouveau rendement pour chaque grande étape de la vie d'un produit, se basant sur différents critères. On s'intéresse ici à l'amélioration du rendement dit "paramétrique". Il s'agit de minimiser l'impact prévisible des

fluctuations de la technologie sur les performances du MEMS en jouant sur les dimensions des masques. On pourrait nommer ce dernier "rendement de conception", puisqu'il se situe en amont de la fabrication.

### a) Défauts catastrophiques et paramétriques

Pour une technologie bien établie, les principaux défauts faisant chuter le rendement de fabrication sont catastrophiques [Figure 1-9] ou paramétriques [JIAN99]. Dans le premier cas, le MEMS ne fonctionnera pas du tout : stiction, poussières [GUED90] ou cassures l'empêcheront de bouger correctement. Ces défauts sont associés au rendement dit *fonctionnel*.



**Figure 1-9 : Phénomène de stiction entre un engrenage et une crémaillère**

Dans le second cas, le MEMS fonctionnera, mais en dehors des spécifications : la dérive des paramètres technologiques réels par rapport aux valeurs utilisées lors des simulations en phase de conception en est la cause. On parle ici de rendement *paramétrique*.

### b) Amélioration en amont de la fabrication

Le rendement paramétrique est donc un sous ensemble du rendement de fabrication, et on s'attache à l'optimiser dès la phase de conception des masques. Il nécessite une bonne connaissance de la technologie : connaître les dispersions des paramètres technologiques, ainsi que les éventuelles corrélations existant entre eux, d'où l'aspect statistique de l'optimisation du rendement paramétrique. On peut par la suite s'intéresser à l'optimisation de composants de base mais aussi à des notions plus complexes telles que le placement-routage en vue d'améliorer le rendement [KORE00].

La Maîtrise Statistique des Procédés (MSP ou Statistical Process Control) permet le suivi de l'évolution de la valeur moyenne et de l'écart type des paramètres technologiques

qui sont retranscrits sur des "cartes de contrôle". Cela permet de suivre les déviations éventuelles de la technologie, mais aussi d'obtenir les données statistiques qui seront utilisées ultérieurement lors de l'optimisation paramétrique du rendement pour les MEMS en cours de conception [MICR99], ou pour retoucher les masques existant.

### **c) Amélioration en ligne**

Si l'on dispose des capteurs et des méthodes de mesure adéquats, on peut utiliser la MSP en "temps réel" et modifier certains paramètres de commande pour limiter les dérives de la technologie. L'APC (Advanced Process Control) est le terme qui recouvre l'ensemble des méthodes de la boucle : mesures – traitement des données – modification éventuelle des variables de commande de l'équipement de fabrication. Dans le secteur de la microélectronique, le nombre de paramètres à prendre en considération a accéléré l'automatisation du processus et le passage de la MSP à l'APC. L'informatique permet de considérer les cartes de contrôle dans leur ensemble malgré leur nombre, et de réaliser des analyses multivariées. On peut ainsi mettre en évidence des corrélations entre paramètres et mieux interpréter l'ensemble des mesures. Un autre terme anglais, *run by run control*, est aussi utilisé pour ce type de méthodes [GOOD97] [MOYN95].

On peut aller encore plus loin dans cette voie : la Maîtrise Globale des Processus (MGP ou *Global Process Control*) [MICH00]. La simple étude des cartes de contrôle ne permet pas toujours d'établir de relation de cause à effet entre les variations des paramètres mesurés et l'éventuelle survenue d'une anomalie sur l'un d'entre eux. On définit donc des indicateurs où l'on tient compte de différents paramètres. Ces derniers sont à même de déceler une anomalie non visible sur une seule carte de contrôle. Chaque nouveau type d'anomalie détectée voit ainsi sa signature enregistrée. Lorsqu'en cours de fabrication, la signature d'une anomalie est reconnue, on peut décider d'agir sur les commandes avant de sortir des spécifications, d'arrêter complètement la chaîne ou de continuer la production, chaque anomalie n'ayant pas la même importance ni le même impact en terme de coût.

La qualité des matériels, des équipements et de la salle blanche est prépondérante pour réduire le nombre de défauts catastrophiques et l'importance des défauts paramétriques, mais se focaliser sur ces facteurs est à la fois très onéreux et insuffisant.

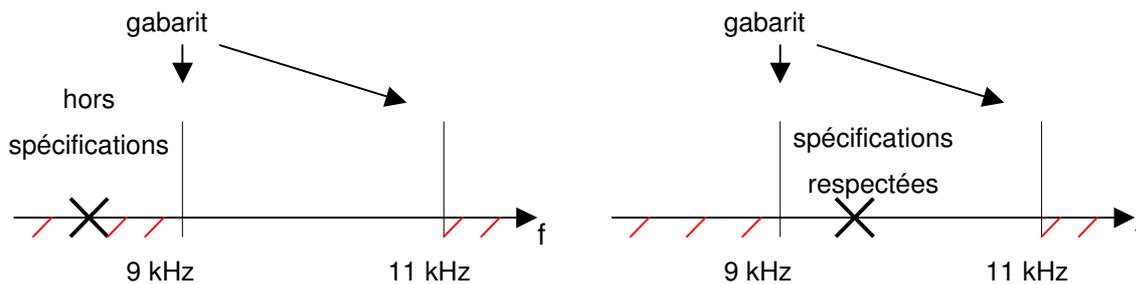
On comprend aisément que le problème du rendement paramétrique doit être pris en considération le plus tôt possible pour limiter les interventions en cours de production, qui sont beaucoup plus complexes et onéreuses.

### 1.2.2 RENDEMENT ET CAO MICROELECTRONIQUE

Depuis les débuts de la CAO pour les MEMS, on s'attache à reproduire le flot de conception tel qu'il existe en microélectronique analogique classique. Une phase d'optimisation nominale est souvent nécessaire en prélude à la phase d'optimisation du rendement, dite statistique. La montée en puissance des microprocesseurs permet l'utilisation, autrefois impossible, d'algorithmes requérant de nombreuses simulations [KELT01].

#### a) Optimisation nominale

La plupart des logiciels de conception comporte au moins un algorithme d'optimisation nominale [BLAN99] (Agilent Technologies en propose onze !). Le concepteur devra avoir des connaissances en optimisation pour choisir l'algorithme en fonction de son problème. L'optimisation statistique étant très fine et gourmande en nombre de simulations, il est important d'avoir de bonnes conditions initiales, c'est à dire que les valeurs initiales des paramètres du circuit assurent déjà des performances intéressantes (par exemple une fréquence de résonance) [Figure 1-10].



**Figure 1-10 : L'optimisation nominale : ajuster les performances d'un système**

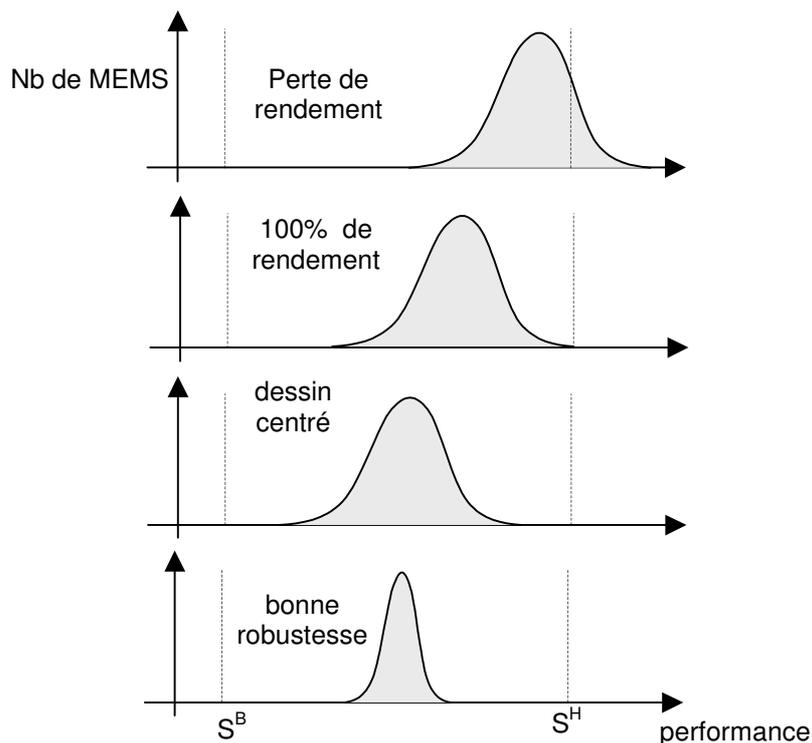
Généralement, les concepteurs de circuits savent quels sont les paramètres critiques sur lesquels jouer. Mais quand ils ont affaire à un nouveau type de circuit comportant de nombreuses valeurs à optimiser, l'optimisation nominale peut être intéressante, et c'est le cas pour les MEMS. Les algorithmes sont très nombreux et variés dans ce domaine [GILB97] [POLA97] [GILL81] : méthodes de type Newton ou quasi-Newton, gradient conjugué, descente de gradient, méthode des régions de confiance, méthode des points intérieurs, algorithmes génétiques [KIRK99]...

#### b) Optimisation du rendement

L'intérêt pour les outils de DFM (*Design For Manufacturability*) va grandissant [FINK99]. La société Cadence propose un module d'optimisation statistique du rendement paramétrique depuis juillet 2002 (Circuit Surfer) [CIRC-net] [GUAR00] [CHEW98]. Le module

de Mentor Graphics (Aspire) est plus ancien [ASPI-net] [DEAL97]; à l'origine, ASPIRE était un projet de ST-Microelectronics, pris en charge par Anacad. Mais l'utilisation en était trop lourde et il n'est plus développé depuis 1996 car très peu vendu. Cela met l'accent sur l'importance de la facilité d'utilisation d'un produit : même si celui-ci est critique, son utilisation ne dépendra que du bon vouloir des concepteurs. Il est donc important de le rendre attractif : par son interface évidemment, mais aussi dans la mesure du possible en proposant une saisie du problème qui limite les connaissances requises en mathématique, en optimisation ou en statistique, du concepteur. Dans le cadre des circuits discrets ou intégrés, on peut chercher à réduire directement les tolérances, mais cela concerne le technologue et non le concepteur de MEMS.

La finalité de ces outils est non seulement d'améliorer le rendement mais aussi d'aller plus loin en améliorant la *robustesse* des produits, c'est pourquoi on parle parfois d'aller au-delà d'un rendement de 100%. Suite à l'optimisation nominale, les spécifications semblent respectées. Mais si l'on considère l'incertitude qu'il existe sur les paramètres technologiques, une analyse de Monte-Carlo peut révéler rapidement que de nombreux MEMS sont hors spécifications et font chuter le rendement [Figure 1-11] ("S<sup>B</sup> et S<sup>H</sup>" sont utilisés pour "spécifications basses et hautes"). L'optimisation statistique nous permettra d'améliorer ce rendement et si possible d'augmenter la robustesse des MEMS en minimisant la variabilité des performances intéressantes et en éloignant leurs valeurs moyennes des valeurs limites [MUCH02]. Cela permet de réduire l'importance des défauts paramétriques qui surviennent parfois dans une chaîne de fabrication sans que la source en soit aisément décelable. Si les MEMS sont conçus de manière robuste, il sera peut être possible de traiter le problème sans arrêter la production et donc sans perte financière. Cela permet aussi d'espérer un fonctionnement normal malgré des variations des variables environnementales (température, tension d'alimentation...) plus importantes que spécifiées.

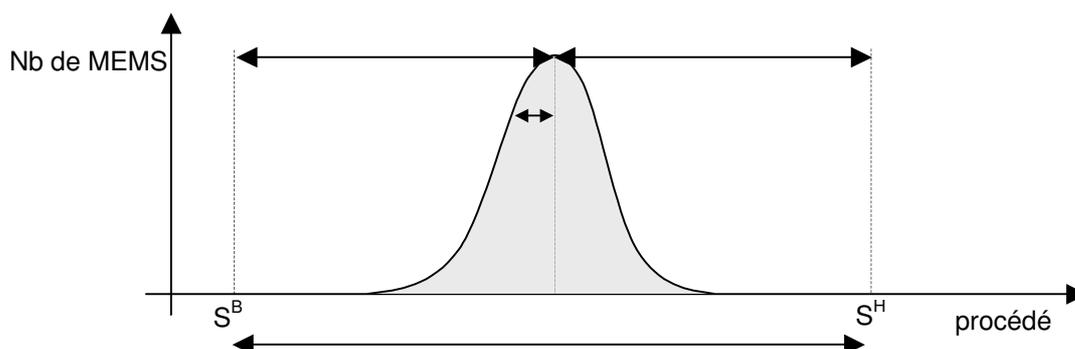


**Figure 1-11 : Amélioration du rendement et de la robustesse**

### 1.2.3 RENDEMENT ET INDUSTRIE

La qualité et le rendement sont très liés au sein d'une entreprise [COME00]. Les méthodes parfois employées dans des algorithmes trouvent leur pendant dans le domaine du management. On parle souvent de l'approche de Taguchi [PILL01] ou de la méthode des à identifier, en un minimum d'expériences (ou simulations), les paramètres critiques d'un processus de fabrication ou de tout autre processus (organisation hiérarchique d'un service...), et de l'optimiser [UNAL91]. La compétitivité et le fort rendement de l'industrie japonaise dans les années 60 et 70, en regard à l'industrie occidentale, est due à l'utilisation de telles méthodes, favorisant l'amélioration des produits ou processus en se basant sur l'expérience des personnels impliqués autant que sur des essais en production où le processus est considéré comme une boîte noire dont on désire améliorer le fonctionnement à moindre coût. L'approche occidentale a longtemps été moins empirique, et beaucoup plus d'efforts étaient consacrés en premier plan à la compréhension des phénomènes pour améliorer le rendement. Mais pour faire face à la prise de parts de marché de plus en plus importante de l'industrie japonaise au cours des années 70, les Etats Unis ont du revoir leur approche de la Qualité. Pendant les années 80, ils se convertissent à l'approche japonaise; dans les années 90, c'est l' Europe qui suit le même chemin.

La méthode des  $6\sigma$  est aussi un outil utilisé dans une approche Qualité Totale, dont le but est la maîtrise des dispersions [Figure 1-12] afin d'accroître les performances financières. Elle a été définie en 1985 par Motorola, et de plus en plus de grandes entreprises s'y rallient (General Electric, Nokia, Sony, Polaroid, Toshiba, Ford Motor...) [LURE]. Elle consiste essentiellement à réduire suffisamment les tolérances des procédés de fabrication pour que toutes les pièces fabriquées soient dans les spécifications.



**Figure 1-12 : Faible dispersion d'un procédé de fabrication**

On estime que le coût de la non-qualité (produits hors spécifications) atteint 15 à 30% du chiffre d'affaire des entreprises. L'atteinte du  $6\sigma$  permettrait de réduire ce coût à moins de 10%.

Nous avons vu dans ce chapitre que les microsystèmes requièrent une technologie complexe mais que certaines applications en sont déjà au stade industriel. Les outils de CAO, et notamment ceux d'optimisation du rendement, sont absolument nécessaires au bon développement de cette nouvelle branche du marché des semiconducteurs, tout comme ce fut le cas pour les circuits intégrés.

## REFERENCES

- [ABAQ-net] <http://www.aceng.com/>
- [AD-net] <http://www.analog.com/technology/mems/>
- [AGIL-net] <http://eesof.tm.agilent.com/docs/adsdoc2002C/optstat/index.html>
- [ALCA-net] <http://www.alcatel.com/telecom/optronics/include/index.html>
- [ANSO-net] <http://www.ansoft.com/products/hf/hfss/>
- [ANSY-net] <http://www.ansys.com/ansys/mems/>
- [ASPI-net] <http://www.mentor.com/ams/dfm.html#aspire>
- [BLAN99] J.P.Blanc et al., "Automatic evaluation of sensors geometrical parameters", Part of the symposium on Design, Test, and microfabrication of MEMS and MOEMS, March 1999, Paris, SPIE Vol. 3680, pp 94-102.
- [CFDR-net] <http://www.cfdrc.com/datab/Applications/MEMS/mems.html>
- [CHAN99] Edward Keat Leem Chan, "Characterization and modeling of electrostatically actuated polysilicon micromechanical devices", Th. : Elec. : Stanford University : 1999 : 195 p.
- [CHEW98] Marko Chew, Sharad Saxena, Thomas F. Cobourn, P.K. Mozumder, Andrzej J. Strojwas, "A new methodology for concurrent technology development and cell library optimization", 12th International Conference on VLSI Design - "VLSI for the Information Appliance" IEEE, Janvier 1999, Goa, India 1998.
- [CIRC-net] [http://www.pdf.com/services\\_tech.phtml](http://www.pdf.com/services_tech.phtml) et [http://www.cadence.com/company/pr/031301\\_PDF.html](http://www.cadence.com/company/pr/031301_PDF.html)
- [CODE-net] <http://www.opticalres.com/cv/cvprodds.html>
- [COME00] A.R. Comeau, "Mixed-signal yield improvement, the human factor", journal : 12<sup>ème</sup> édition de Semiconductor Fabtech, Juillet 2000, 303 p. Disponible sur <http://www.fabtech.org/journals/edition.12/section2.shtml>.
- [COVE-net] <http://www.coventor.com/>
- [DEAL97] M. De Almeida et al., "A statistical method for the analysis of CMOS process fluctuations on dynamic performance", Proceedings IEEE 1997, International Conference on Microelectronic Test Structures, Vol. 10, Monterey, U.S.A..
- [FINK99] S. Finkbeiner et al., "Simulation of nonideal behaviour in integrated piezoresistive silicon pressure sensor", Part of the symposium on Design, Test, and microfabrication of MEMS and MOEMS, March 1999, Paris, SPIE Vol. 3680, pp 188-199.

- [GADE01] Mohamed Gad-El-Hak, "The Mems Handbook", septembre 2001, CRC Press, 928 p.
- [GILB97] J.C. Gilbert, J.F. Bonnans, C. Lemaréchal et C.A. Sagastizábal, "*Optimisation Numérique: Aspects Théoriques et Pratiques*", Ed. Springer, 1997.
- [GILL81] P.E. Gill, W. Murray and M.H. Wright, "Practical Optimization", Academic Press, Londres, 1981.
- [GOOD97] Brian Goodlin, "Run by Run control of endpoint uniformity in a plasma etching process", Th. proposal : Elec. : Massachusetts Institute of Technology : 1997 : 24 p.
- [GUAR00] C. Guardini et al., "An asymptotically constant, linearly bounded methodology for the statistical simulation of analog circuits including component mismatch effects", DAC 2000, Los Angeles, California.
- [GUED90] David Guedj, "Validation d'un modèle de rendement. Application à l'optimisation du dessin de circuits intégrés", Th. : Elec. : Université de Paris Sud, Centre d'Orsay : 1990 : 228 p.
- [GUPT97] Raj K. Gupta, "Electrostatic pull-in test structure design for in-situ mechanical property measurement of microelectromechanical systems (MEMS)", Th. : Elec. : Massachusetts Institute of Technology : 1997 : 68 p.
- [HECT99] Héctor J. De Los Santos, "Introduction to electromechanical microwave systems", Artech House, 1999, 215 p.
- [HP-net] <http://www.hp.com/oeminkjet/evaluate/index.htm>
- [INTE-net] <http://www.intellisense.com/home.asp>
- [JIAN99] T. Jiang, C. Kellon and R.D. Blanton, "Inductive Fault Analysis of a Microresonator", Technical Proceedings of the 1999 International Conference on Modeling and Simulation of Microsystems, San Juan, Puerto Rico, pp. 498 – 501
- [JONE99] P.T. Jones, G.C. Johnson, R.T. Howe, "Statistical characterization of fracture of brittle MEMS Material", SPIE Vol. 3880, September 1999, Santa Clara, California, Part of the SPIE Conference on MEMS Reliability for Critical and Space Applications, pp 20-29.
- [KELT01] W.D. Kelton, "Some modest proposals for simulation software: design and analysis of experiments", Proceedings of the 34th Annual Simulation Symposium (part of the Advanced Simulation Technologies Conference), Seattle, 2001, 6 p.
- [KIRK99] G.A. Kirkos et al. "MEMS optimization incorporating genetic algorithms", Part of the symposium on Design, Test, and microfabrication of MEMS and MOEMS, March 1999, Paris, SPIE Vol. 3680, pp 84-93.

- [KORE00] I. Koren, "Should yield be a Design objective ?", First International Symposium on Quality of Electronic Design, IEEE, March 2000, San Jose, California, pp. 115-120.
- [KRUG99] Ezekiel John Joseph Kruglick, "Microrelay design, performance, and systems", Th. : Elec. : University of California, Berkeley : 1999 : 219 p.
- [LAFO01] Xavier Lafontan, "Application des micro-technologies aux composants de commutation électrique : études et performances de micro-relais", Th. : Elec. : Université de Montpellier II : 2001 : 203 p.
- [LATO99] Laurent Latorre, "Evaluation des techniques microélectroniques contribuant à la réalisation de microsystèmes : application à la mesure du champ magnétique", Th. : Elec. : Université Montpellier II : 1999 : 138 p.
- [LORE99] G. Lorenz, R. Neul and S. Dickmann, "Modeling of electrostatic MEMS components", Technical Proceedings of the 1999 International Conference on Modeling and Simulation of Microsystems, San Juan, pp.128-131.
- [LURE] Séminaire LURE (Laboratoire pour l'Utilisation du Rayonnement Electromagnétique), "Le déploiement de la politique qualité groupe", disponible sur : <http://www.lure.u-psud.fr/Seminaires/documents/Framatome/lure6.pdf>
- [MADO02] Marc J. Madou, "Fundamentals of Microfabrication: The Science of Miniaturization", 2nd édition (avril 2002) CRC Press, 608 p.
- [MAST99] C.H. Mastrangelo, "Suppression of stiction in MEMS," invited paper, 1999 Spring MRS Meeting, Boston.
- [MCCL99] F.P. McCluskey, M. Pecht, "Rapid reliability assessment using CADMAP-II", proceedings of Modeling and Simulation of Micosystems 1999, San Juan, USA, disponible sur <http://www.cr.org>.
- [MEMS-net] <http://www.memscap.com>
- [MICH00] Daniel Lafaye de Micheaux, "Prolonger la MSP par la maîtrise globale d processus", revue "Qualité Références", juillet 2000, pp. 47-55.
- [MICR99] F. Bergeret, Y. Chandon, "Improving Yield in IC manufacturing by statistical analysis of a large database", revue "Micro", mars 1999, pp. 59-75.
- [MOUL02] D. Moulinier, P. Nachtergaele, M-P. Brutails, S. Bergeon, S. Maza, Ch. Hui Bon Hoa, "Efficient MEMS design methodology integrating finite elements based macro models into a nodal approach MEMS component design tool", Proceedings of Design, Test, Integration and Packaging of MEMS/MOEMS : DTIP 2002, Cannes, 9 p.
- [MOYN95] William P. Moyne, " Run by Run control: interfaces, implementation, and integration", MS Th. : Elec. : Massachusetts Institute of Technology : 1995 : 87 p.

- [MUCH02] J. Muchow et al., "Influence of process variation on the functionality of a high pressure sensor", Design, Test, and microfabrication of MEMS/MOEMS 2002, Proceedings of SPIE Vol. 4755, pp 143-148.
- [MUKH99] N. Deb, S. V. Iyer, T. Mukherjee and R.D.S. Blanton, "MEMS Resonator Synthesis for Testability", Symposium on Design, Test, and Microfabrication of MEMS/MOEMS 1999, Paris, pp. 58-69.
- [MUMP] MUMPs™ Design Handbook, version 7.0, disponible sur <http://www.memsrus.com>.
- [NAGL98] O. Nagler, M. Trost, B. Hillerich and F. Kozlowski, "Efficient design and optimization of MEMS by integrating commercial simulation tools", Sensors and Actuators A: Physical, Volume 66, Issues 1-3, 1 April 1998, pp. 15-20.
- [NATH67] H. Nathanson et al., "The Resonant Gate Transistor", Mar. 1967, Transactions on Electron Devices, vol. ED-14 No. 3, IEEE, pp. 117-133.
- [NGUY98] C.T.-C.Nguyen, "Micromachining technologies for miniaturized communication devices", SPIE conference on Micromachined Devices and Components IV, 1998, Santa Clara U.S.A., pp. 24-38.
- [OPTI-net] <http://www.optiwave.com/>
- [PELE98] Philippe Pelegri, "Microsystèmes et micropompes", rapport de stage-DESS réalisé au CNES-Toulouse en 1998.
- [PILL01] Maurice Pillet, "Les plans d'expériences par la méthode Taguchi", 3e tirage 2001, Paris : Ed. d'Organisation, 330 p.
- [POLA97] E. Polak, "Optimization : Algorithms and Consistent Approximations", Springer, New York, 1997 VI, 779 p.
- [RSOF-net] <http://www.rsoftinc.com/products.htm>
- [SANC00] Susan M. Sanchez, "Robust design: seeking the best of all possible worlds", Proceedings of the 2000 Winter Simulation Conference, ed. J.A. Joines et al., pp. 69-76.
- [SAND-net] <http://www.mdl.sandia.gov/>
- [TI-net] <http://www.dlp.com>
- [UNAL91] R. Unal, E. B. Dean, "Design for cost and quality: the robust design approach", Journal of Parametrics, Vol. XI, No. 1, août 1991.
- [WACH99] Gerhard Wachutka, "The art of modeling coupled field effects in microdevices and microsystems", proceedings of Modeling and Simulation of Micosystems 1999, San Juan, U.S.A., disponible sur <http://www.cr.org>.
- [ZHAN98] Gang Zhang, "Design and simulation of a CMOS-MEMS accelerometer", MS Th. : Elec. : Tsinghua University : 1998 : 39 p.

## Chapitre 2

# OPTIMISATION STATISTIQUE DU RENDEMENT PARAMETRIQUE



## 2. OPTIMISATION STATISTIQUE DU RENDEMENT PARAMETRIQUE

Ce chapitre entre dans le vif du sujet : après avoir repositionné les données du problème (performances, paramètres impliqués, flot d'optimisation...), nous faisons un état de l'art des méthodes classiques d'estimation et d'optimisation du rendement. Nous donnons notamment des notions élémentaires pour manipuler à bon escient des plans d'expériences simples.

### 2.1 RENDEMENT : DEFINITIONS

Tous les paramètres technologiques sont sujets à des fluctuations. Pour cette raison, on leur confère un aspect statistique (valeur moyenne et dispersion dans le cas d'une représentation gaussienne). Il faut éviter que ces variations inopinées puissent faire sortir un MEMS des spécifications qui lui sont imposées. Ces fautes sont liées au rendement dit paramétrique :

$$\text{rendement paramétrique} = \frac{\text{nombre de MEMS fonctionnant dans les spécifications}}{\text{nombre de MEMS fonctionnant fabriqués}}$$

alors que les défauts catastrophiques entraînent une chute du rendement fonctionnel :

$$\text{rendement fonctionnel} = \frac{\text{nombre de MEMS fonctionnant}}{\text{nombre de MEMS fabriqués}}$$

Par la suite, nous parlerons simplement de "rendement" pour rendement paramétrique. Par ailleurs, nous utiliserons souvent le terme *réponse* dans le sens de *performance* du MEMS, et nous parlerons indifféremment de réduction de la dispersion, de l'écart type, de la variabilité ou de la variance, implicitement en référence à une performance.

#### 2.1.1 LES PERFORMANCES CIBLEES

Elles peuvent se présenter sous différentes formes : courbes cibles (*targets*), limites maximum ou minimum ("spécifications"), gabarit formant une enveloppe, fonction mathématique... Dans tous les cas, pour faciliter le processus d'optimisation, il est important de réduire ces contraintes au strict minimum. Dans le cas des limites, cela revient par exemple à en limiter le nombre ou à les fixer le plus largement possible dans un premier temps.

### a) Courbes cibles

Il s'agit d'indiquer l'allure idéale que devrait suivre la courbe retraçant la réponse d'un MEMS, par exemple lors d'une analyse AC [Figure 2-1]. La cible peut être définie par morceaux.

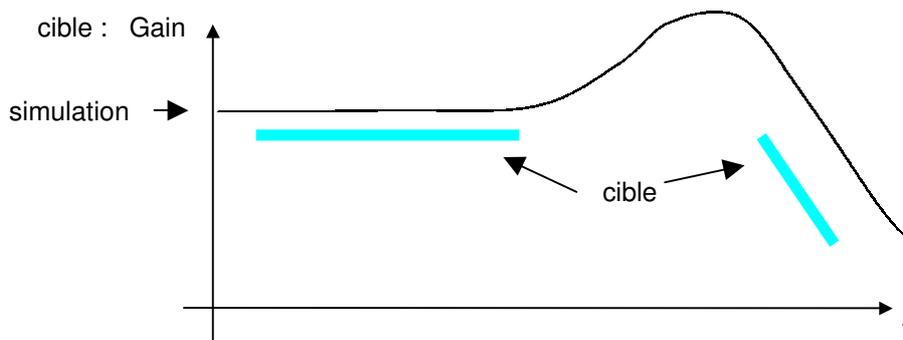


Figure 2-1 : Analyse AC d'un MEMS et sa courbe cible associée

### b) Limites et gabarits

Comme nous avons pu déterminer un but vers lequel tendre, nous pouvons déterminer des zones dangereuses à éviter : les limitations supérieures et inférieures [Figure 2-2]. Lorsque les deux types de limite sont considérés pour un même point de la réponse du système, ils délimitent un gabarit.

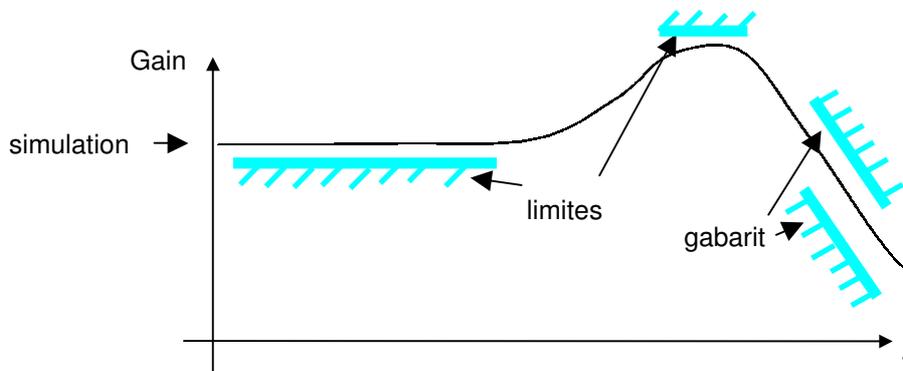


Figure 2-2 : Analyse AC d'un MEMS et limites associées

### c) Fonctions mathématiques

Dans les cas précédents, nous considérons des vecteurs. Mais nous pouvons aussi considérer des scalaires tels qu'une fréquence de résonance, un déplacement maximal, une bande passante, etc. On traite le vecteur (résultat de l'analyse désirée) pour en extraire l'information scalaire qui nous intéresse. Pour cela, on utilise des fonctions mathématiques de type  $\min()$ ,  $\max()$  ou  $20 \cdot \log()$ , mais on peut en imaginer de plus complexes.

### 2.1.2 LE FLOT D'OPTIMISATION

L'optimisation du rendement s'inscrit dans un flot global [Figure 2-3] qui regroupe différentes techniques. Suivant la qualité du circuit initial, les plus rudimentaires ne seront pas utilisées. Si le concepteur a affaire à un circuit auquel il n'est pas familier, il peut avoir recours à une analyse de sensibilité. Cette dernière lui donnera une information quant à la criticité des paramètres. Les paramètres sensibles étant identifiés, une analyse paramétrique permettra d'approcher des valeurs optimales. Si le nombre de paramètres est élevé, on peut souhaiter utiliser une optimisation nominale qui automatise cette démarche. Enfin, ce n'est qu'en dernier lieu, une fois que les conditions initiales sont bonnes, que l'on fait appel à l'optimisation statistique.

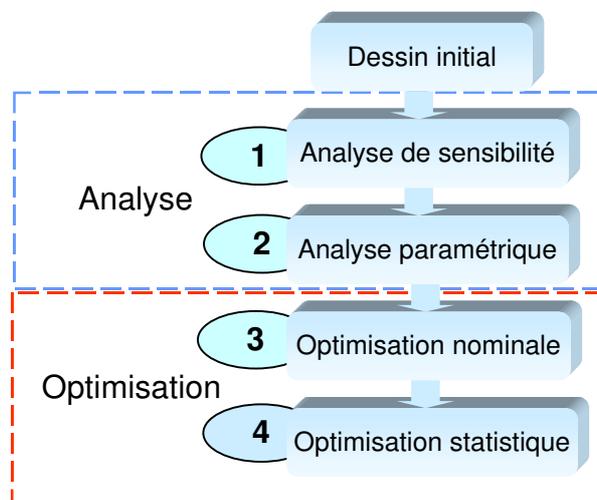


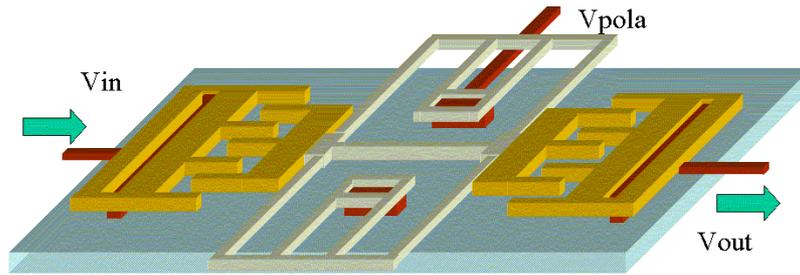
Figure 2-3 : Flot d'optimisation des performances d'un MEMS

## 2.2 DIFFERENTES SORTES DE PARAMETRES

Les paramètres de conception présents dans les netlists se regroupent en 3 types : les paramètres de dessin, les paramètres technologiques et enfin les paramètres mixtes, qui possèdent les deux aspects.

### 2.2.1 PARAMETRES DE DESSIN

Prenons l'exemple d'un MEMS classique : une structure composée de 2 peignes électrostatiques et d'une masse, supportée par des poutres faisant office de ressorts [Figure 2-4]. Cette structure est souvent utilisée comme filtre ou résonateur. Concernant la partie mécanique, la fréquence de résonance propre est notamment fonction de ses dimensions géométriques. La partie électrostatique, elle, sert à actionner la structure et à capter l'amplitude de son déplacement.



**Figure 2-4 : Structure à peignes électrostatiques**

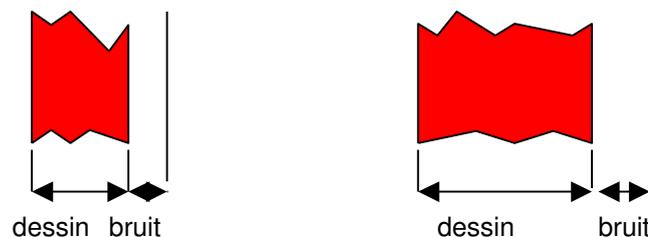
Les paramètres de dessin (ou "de conception") sont ceux sur lesquels le concepteur peut jouer pendant la phase d'optimisation : le nombre de doigts des peignes, le nombre de spires des ressorts, les dimensions géométriques...

### 2.2.2 PARAMETRES TECHNOLOGIQUES

Ils sont parfois appelés paramètres de bruit, et on leur attache un aspect statistique. C'est le cas par exemple du module d'Young, qui intervient dans l'estimation des déplacements de toutes les parties mécaniques des MEMS, notamment dans le calcul des constantes de raideur des ressorts. La résistivité, la densité des matériaux, l'épaisseur des différentes couches de la technologie sont autant de paramètres sur lequel le concepteur ne peut intervenir. En revanche, pour une optimisation statistique, il est nécessaire que le fondeur puisse fournir leur dispersion. Cela peut être problématique, car il s'agit de données très sensibles pour ce dernier.

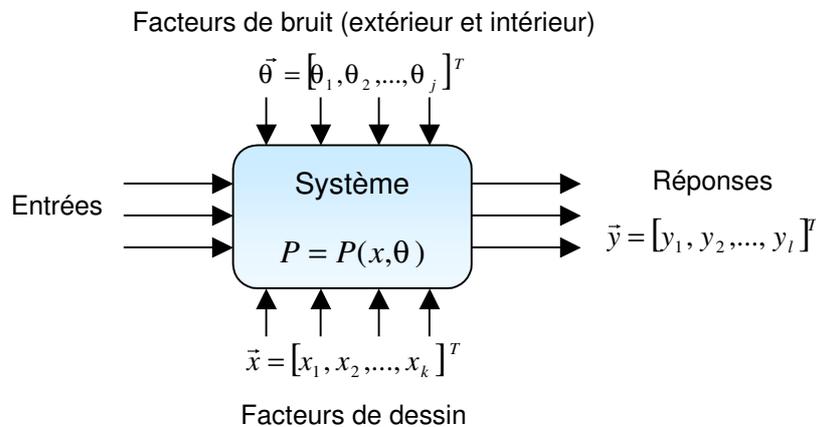
### 2.2.3 PARAMETRES MIXTES

Les paramètres les plus nombreux sont mixtes. Ce sont des paramètres de dessin, mais qui possèdent un aspect statistique qui en fait aussi des paramètres de bruit [Figure 2-5]. Il s'agit tout simplement de toutes les dimensions géométriques d'une structure qui peuvent apparaître sous forme de paramètres dans la netlist : largeur d'une piste, longueur d'une poutre...



**Figure 2-5 : La largeur d'une piste est un paramètre mixte**

L'ensemble des paramètres étant définis, on peut normaliser la notation que l'on utilisera par la suite. Le concepteur désire manipuler  $k$  paramètres de dessin au cours de l'optimisation, notés  $x_i$  et représentés par le vecteur  $\vec{x} = [x_1, x_2, \dots, x_k]^T$ . De même il existe un vecteur correspondant aux variables aléatoires attribuées à chaque paramètre de bruit  $\vec{\theta} = [\theta_1, \theta_2, \dots, \theta_j]^T$ , ainsi qu'un vecteur représentant les performances obtenues après simulation du microsystème :  $\vec{y} = [y_1, y_2, \dots, y_l]^T$ .  $P$  représente le vecteur des paramètres modifiés dans la netlist, on a donc  $P = P(x, \theta)$ . Le schéma global est représenté sur la Figure 2-6.

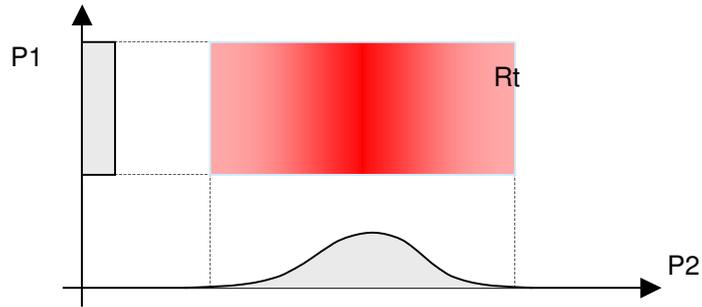


**Figure 2-6 : Représentation des variables, données de l'optimisation**

Certains paramètres "extérieurs" considérés comme des variables environnementales (bruit extérieur) tels que la tension d'alimentation peuvent être traités comme des paramètres mixtes, et la température comme un paramètre technologique [POTT01].

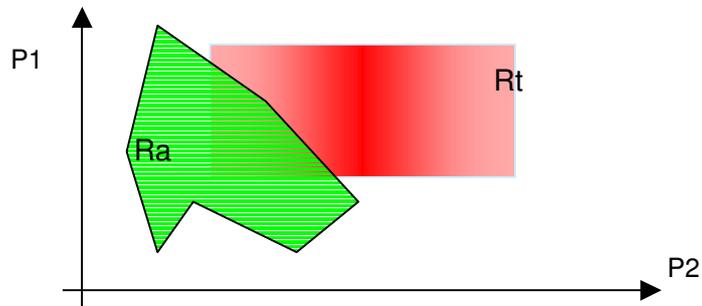
#### 2.2.4 PARAMETRES ET ESPACES VECTORIELS

Le problème posé par l'optimisation statistique est souvent considéré d'un point de vue vectoriel. Soit un espace vectoriel dont la base est l'ensemble des paramètres considérés pour l'estimation ou l'optimisation du rendement. On nomme Région de Tolérance (Rt) la zone de cet espace formée par l'ensemble des valeurs que pourraient prendre ces paramètres compte tenu de leur dispersion [Figure 2-7].



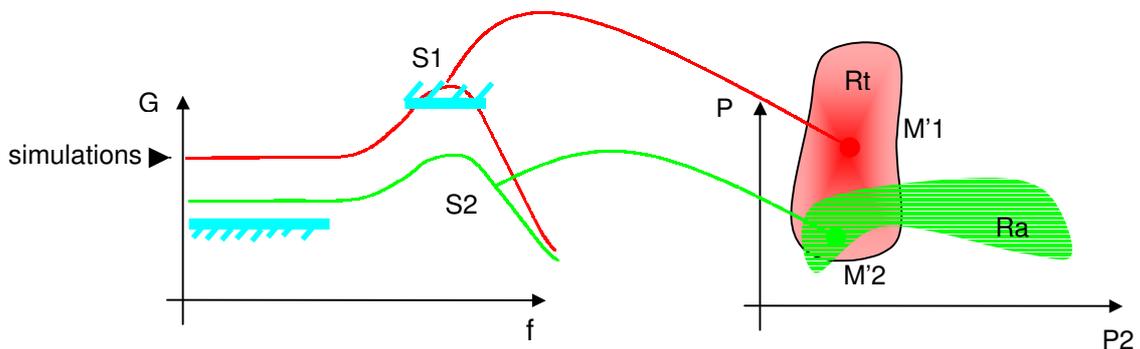
**Figure 2-7 : Région de Tolérance (Rt)**

Une deuxième zone de cet espace est remarquable : la Région d'Acceptabilité (Ra). Cette dernière est a priori initialement inconnue : c'est la zone où les paramètres ont des valeurs qui permettent au microsystème de respecter les spécifications imposées [Figure 2-8].



**Figure 2-8 : Région d'Acceptabilité (Ra)**

Optimiser le rendement revient finalement à jouer uniquement sur les paramètres de dessin pour déplacer Rt et tendre vers un point où le recouvrement de Rt par Ra est plus important, voire total [Figure 2-9], ce qui signifie que tous les points de la région de tolérance sont de bonnes solutions pour respecter les spécifications.



**Figure 2-9 : Recouvrement de Ra et Rt pour un MEMS dans les spécifications**

Par contre, dans un processus d'estimation, on cherche seulement à déterminer l'ampleur de ce recouvrement. Lorsque ce dernier est partiel et que les dispersions des paramètres de bruit ne sont pas uniformes, le pourcentage de la surface recouverte n'est

pas directement l'image du rendement : il faut découper cette surface et pondérer ses composantes en faisant intervenir des coefficients pour les paramètres dont les distributions ne sont pas uniformes.

### 2.3 ESTIMATION DU RENDEMENT

Les méthodes d'estimation du rendement sont très nombreuses, et chacune peut comporter plusieurs variantes. Elles peuvent être classées suivant de nombreux critères. L'estimation du rendement revient à réaliser une analyse statistique d'un MEMS. Généralement, une variante de l'analyse de Monte-Carlo est utilisée en adéquation avec la méthode choisie. Nous classons les méthodes d'estimation du rendement en 3 branches principales [Figure 2-10].

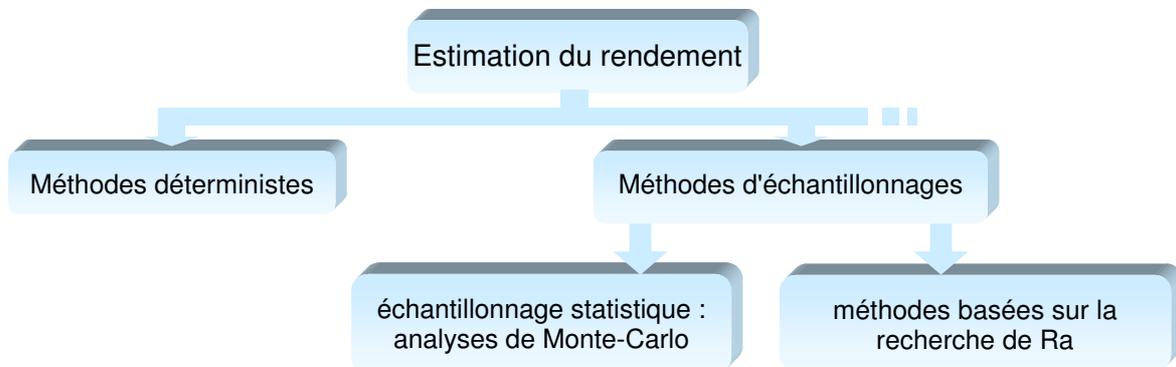


Figure 2-10 : Les différents types de méthodes d'estimation du rendement

#### 2.3.1 METHODES DETERMINISTES

L'approche déterministe utilise des méthodes généralement rapides pour estimer les variations dans les fonctions décrivant une performance (*fonctions de performances*) [ZEBB96]. Ses inconvénients dans notre cas sont de taille : le nombre de paramètres doit être faible et Ra doit être connexe et sans "trous noirs" (volumes cernés par Ra mais ne lui appartenant pas). Dans la littérature, on considère que cinq ou six paramètres reste un nombre acceptable. Au-delà, le nombre de simulations nécessaires semble être prohibitif, il n'est toutefois pas donné dans les publications. Mais nous désirons pouvoir traiter un grand nombre de paramètres, et ne faire aucune hypothèse sur la région d'acceptabilité Ra. Nous nous écarterons donc de cette voie. Notons tout de même que la méthode la plus citée dans la littérature est la *méthode des moments* [ZEBB96] [BRAH84]. Elle est basée sur les formules de transmission des moments, qui découlent d'un développement en série de Taylor de la fonction de performance.

### 2.3.2 ECHANTILLONNAGE STATISTIQUE

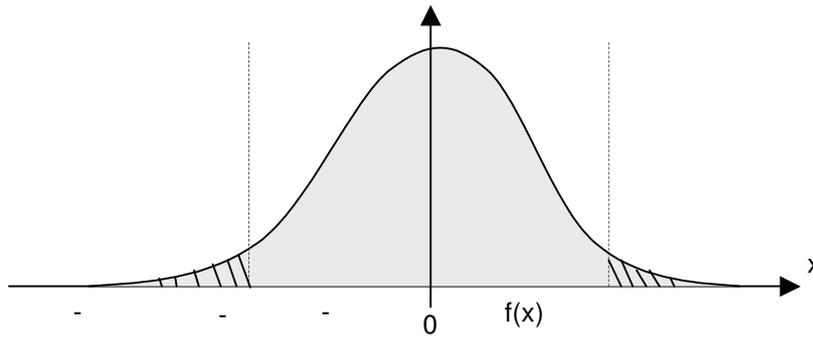
La méthode de Monte-Carlo a fait l'objet de multiples publications, mais sa définition reste imprécise dans la littérature car le même mot est utilisé pour différentes variantes et implémentations dans différents domaines (mathématiques, physique, informatique...). En informatique, elle comprend tout ce qui a trait à l'utilisation de séquences pseudo-aléatoires, utilisées pour résoudre les problèmes numériques qualifiés de *déterministes* tels que calcul d'intégrales, résolution de systèmes, d'équations différentielles ou **d'optimisation**. Il existe bien sûr des méthodes numériques déterministes dédiées à ces problèmes, mais leur complexité est fonction du nombre de paramètres. Une méthode de Monte-Carlo s'appliquera à un problème **sans contraintes sur le nombre de paramètres concernés par l'optimisation, ni même sur leur distribution ou sur Ra**. Ce sont ses grands avantages sur les autres méthodes, nous les verrons par la suite. La perte se trouve dans la précision de la solution, mais peut être réduite en augmentant le nombre de simulations toléré, donc le temps de résolution du problème. Monte-Carlo se retrouve dans presque toutes les méthodes d'estimation ou d'optimisation du rendement : soit c'est la base de l'algorithme, soit on y a recours dans des routines annexes.

#### a) Les lois de probabilité

Les paramètres de bruit sont susceptibles de suivre différentes lois de probabilité : la loi de poisson, la loi normale (de forme gaussienne), etc. Dans notre étude, nous nous limiterons à ce cas le plus général. De même, nous considérons que les performances des MEMS suivent aussi une telle loi. La densité de probabilité de la loi normale est

$$f(x) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{1}{2} \left( \frac{x-\mu}{\sigma} \right)^2} \quad (\text{équ. 2-1}) \quad \text{où } \mu \text{ est la valeur moyenne de la distribution et } \sigma$$

son écart type (ou "déviatation standard"). On peut intégrer cette équation pour connaître la probabilité de tomber dans un intervalle donné [Figure 2-11]. L'intégrale d'une densité de probabilité sur un intervalle donne la probabilité que la variable aléatoire dont elle décrit la distribution prenne une valeur de cet intervalle. Sur l'intervalle  $[-\infty, +\infty]$ , elle est donc égale à 1. La fonction de répartition  $F(x)$  est l'intégrale de la densité de probabilité sur  $[-\infty, x]$ .



**Figure 2-11 : L'intégrale d'une gaussienne sur l'intervalle  $[-n\sigma, n\sigma]$  - représente 95,4% de l'intégrale sur  $[-\infty, +\infty]$**

Le Tableau 2-1 donne la probabilité que cette variable prenne une valeur de l'intervalle  $\mu \pm n\sigma$ . Elle est obtenue par l'intégrale de  $f(x)$  dans cet intervalle.

| n  | 0 | 0,5 | 1  | 1,5 | 2    | 2,5  | 3     |
|--|---|-----|----|-----|------|------|-------|
| valeur de l'intégrale sur $[-n\sigma, n\sigma]$ (en %) | 0 | 38  | 69 | 87  | 95,5 | 98,8 | 99,73 |

**Tableau 2-1 : Valeur de l'intégrale en fonction de l'intervalle considéré**

### b) Interprétation de l'analyse de Monte-Carlo

Certaines méthodes de type Monte-Carlo ont pour but de contrôler (réduire) la variance de *l'estimateur de Monte-Carlo*. Il ne s'agit pas de la variance de la performance du MEMS (que l'on souhaite aussi réduire), mais celle qui retrace l'incertitude du rendement estimé en regard du rendement réel. Les plus communes sont les méthodes d'échantillonnage stratifié ou pondéré [KERAA98], les carrés latins (LHS) [KERAA97]. Mais pour obtenir une réduction de variance intéressante vis-à-vis de l'échantillonnage primitif, il faut faire des hypothèses sur certains caractères de la performance visée, ce qui revient à considérer que le concepteur ait déjà une bonne connaissance de son circuit et des méthodes de type Monte-Carlo, puisqu'il doit être capable de choisir la plus efficace dans son cas.

Evidemment, on ne peut demander à un concepteur de choisir quelle méthode statistique est la mieux adaptée au MEMS qu'il met au point, le domaine ne compte pas encore assez d'experts pour rendre un tel outil intéressant. Pour le choix de l'algorithme d'optimisation, comme pour le choix de l'analyse de Monte-Carlo, nous avons donc décidé d'opter pour des méthodes non optimisées pour un problème donné, mais très robustes quelle que soit la situation. C'est donc l'échantillonnage primitif (PMC) que nous avons retenu, où l'estimateur de Monte-Carlo s'écrit :

$$\hat{Y}_{PMC}(\bar{x}) = \frac{1}{N} \sum_{i=1}^N I_{\theta}(\theta^i, \bar{x}) \text{ et sa variance est } \sigma_{\hat{y}}^2 = \frac{\hat{Y}(1-\hat{Y})}{N} \quad (\text{éq. 2-2}) \quad \text{où } I \text{ est une}$$

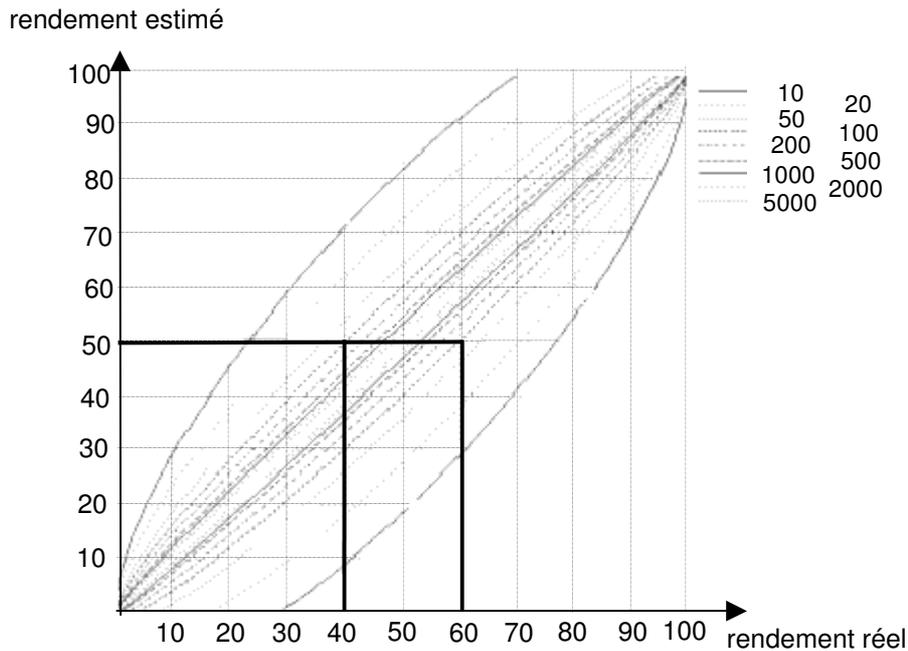
fonction indicatrice telle que:  $I_p(P) = \begin{cases} 1 & \text{si les spécifications sont respectées} \\ 0 & \text{si les spécifications sont violées} \end{cases}$

et N est le nombre d'échantillons, c'est à dire de simulations dans notre cas.  $\hat{Y}$  est le rendement estimé. Des travaux plus poussés ont été menés pour améliorer ces méthodes d'estimation [KERAA98], mais nous n'avons pu implémenter et tester ces versions faute de temps.

La variance de l'estimateur décroît avec l'augmentation du nombre d'échantillons. Il existe des tables couramment utilisées avec la méthode de Monte-Carlo qui lient  $\hat{Y}$ , N, la marge d'erreur  $\varepsilon$  et le niveau de confiance (en France, on utilise son complément, le "seuil de signification"). La marge d'erreur est l'erreur d'estimation que l'on est prêt à accepter ou contrôler, décision, parce que nous ne pouvons pas assurer que tous les échantillons de la population auront la même erreur. Pour contrôler la précision de l'estimation, nous devons aussi contrôler son niveau de confiance, c'est à dire fixer la probabilité que la marge d'erreur soit effectivement inférieure à z%. On définit ce niveau de confiance par l'intégrale sur  $[\hat{y} - n\sigma_{\hat{y}}, \hat{y} + n\sigma_{\hat{y}}]$  de la densité de probabilité de la loi normale (éq. 2-1) en effectuant les changements de variable appropriés [DUBO87].

Soit C le niveau de confiance exprimé en nombre de déviations standards (pour effectuer cette correspondance, on peut se référer au Tableau 2-1). Il est alors lié à N,  $\hat{Y}$  et par :  $N = \left(\frac{C_{\sigma}}{\varepsilon}\right)^2 \hat{Y}(1-\hat{Y})$ , ou encore  $\varepsilon = C_{\sigma} \sqrt{\frac{\hat{Y}(1-\hat{Y})}{N}}$ . Ainsi l'erreur faite sur l'estimation d'un rendement à 80% réalisée en 1600 simulations, pour un niveau de confiance de 95,4% (+/-2 $\sigma$  dans le Tableau 2-1) est de +/- 2%. Cela signifie qu'il y a 95,4% de chance de vérifier l'inégalité :  $\hat{y}(1-\varepsilon) < Y < \hat{y}(1+\varepsilon)$  soit  $78,4 < Y < 81,6$ .

Des abaques [Figure 2-12] sont couramment utilisées et peuvent être implémentées sous forme de tableau [Tableau 2-2] pour fournir à l'utilisateur d'un logiciel toutes les données relatives à l'estimation demandée.



**Figure 2-12 : Abaque du rendement réel en fonction du rendement estimé et du nombre de simulations considéré**

| Niveau de Confiance : 95%<br>Rendement Réel : 80% |                      |     |      |
|---|----------------------|-----|------|
| +/- %   | Rendement estimé (%) |     | N    |
|   | Min                  | Max |      |
| 1   | 79                   | 81  | 6146 |
| 2   | 78                   | 82  | 1536 |
| 3   | 77                   | 83  | 682  |
| 4   | 76                   | 84  | 384  |
| 5   | 75                   | 85  | 245  |
| 6   | 74                   | 86  | 170  |
| 7   | 73                   | 87  | 125  |
| 8   | 72                   | 88  | 96   |
| 9   | 71                   | 89  | 75   |
| 10  | 70                   | 90  | 61   |

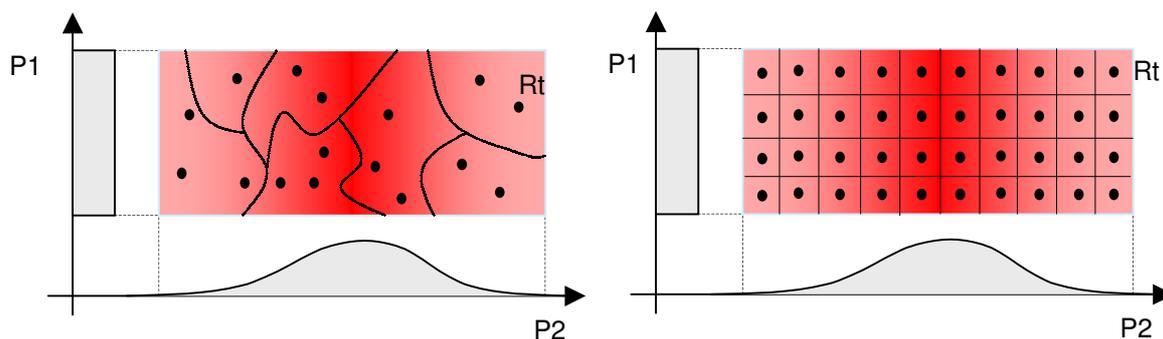
**Tableau 2-2 : Erreur sur le rendement estimé**

### c) Variantes de l'analyse de Monte-Carlo

Pour réduire la variance de l'estimateur de Monte-Carlo sans augmenter le nombre d'échantillons, on peut avoir recours à des méthodes d'échantillonnage différentes. Les plus communes sont les méthodes d'échantillonnage stratifié ou pondéré.

### c.1 Echantillonnage stratifié

Dans cette approche [KERAb98], nous divisons la région de tolérance  $R_t$  en strates [Figure 2-13] et le rendement partiel de chacun de ces sous-espaces est évalué, on peut alors en déduire le rendement global. La *méthode de régionalisation* ressemble à cette approche, mais le découpage y est régulier et le nombre de simulations requises augmente beaucoup plus fortement avec le nombre de paramètres que pour l'échantillonnage stratifié. Il existe de nombreuses variantes qui utilisent différents découpages, notamment en utilisant la méthode des *plans d'expériences* (voir §2.4.3, hypercubes latins...).



**Figure 2-13 : Echantillonnage stratifié et méthode de régionalisation**

### c.2 Echantillonnage pondéré (*importance sampling*)

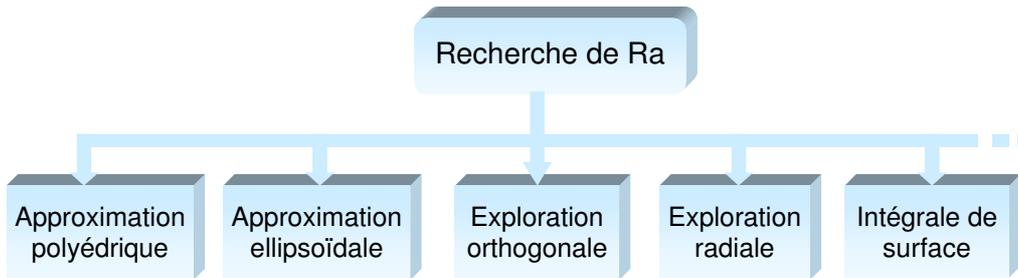
Il s'agit d'une méthode où la variance de l'estimateur n'est pas meilleure que l'estimateur PMC, mais elle peut le devenir si une approximation externe de  $R_a$  est connue, il faut donc réaliser une étape préalable pour cerner  $R_a$ , qui peut s'avérer coûteuse. L'échantillonnage stratifié est plus simple à mettre en place.

### c.3 Plans d'expériences

Les *plans d'expériences* [§2.4.3] permettent de choisir intelligemment les simulations à effectuer pour cerner rapidement l'influence de chaque paramètre du MEMS sur une performance cible. Ils permettent aussi la construction à moindre frais d'une représentation polynomiale de cette performance, par une méthode appelée méthode des surfaces de réponse (*Response Surface Method*) [SMIT98]. A noter que des variantes de cette méthode permettent des représentations différentes, à base de splines ou de fonction de type radial [PETE01]. Suite à la construction d'un polynôme, on peut réaliser une analyse de Monte-Carlo classique sur ce modèle mathématique avec un nombre très important d'échantillons, le temps de simulation n'étant plus une limite [LOW89]. Mais on peut aussi choisir d'utiliser des plans d'expériences directement comme méthode d'échantillonnage [KERA97], de même que l'on peut construire un modèle en utilisant des expériences aléatoires [MILO90].

### 2.3.3 METHODES GEOMETRIQUES

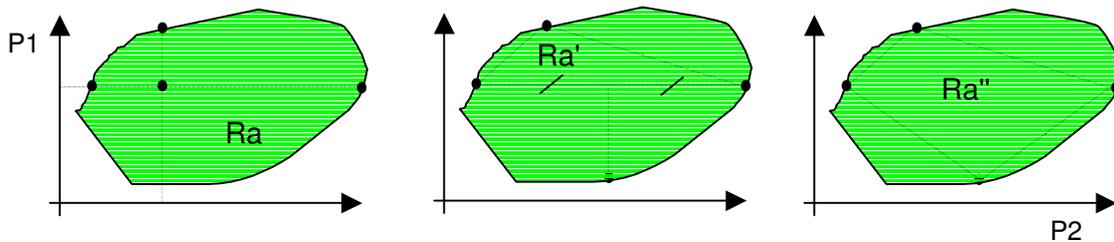
Les tirages aléatoires de valeurs de paramètres peuvent être utilisés de manière "annexe" par des algorithmes déterministes. Basées sur la recherche de Ra, ces approches [Figure 2-14] sont aussi dites *géométriques*. Nous devrions donc parler dans ce cas plus d'amélioration que d'optimisation du rendement. Nous donnons les bases pour cerner la logique des principales méthodes.



**Figure 2-14 : Méthodes d'estimation du rendement basées sur la recherche de la région d'acceptabilité Ra**

#### a) Approximation polyédrique (*simplicial approximation*)

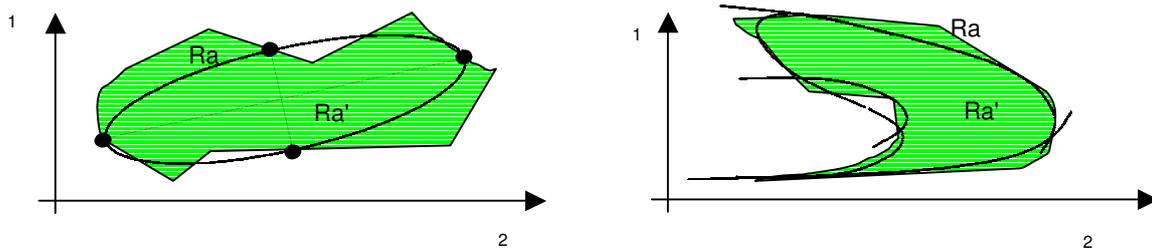
Pour utiliser cette méthode [KERAA98] [ZEBB96] [BRAH84], il faut disposer d'un point appartenant à Ra. Partant de ce point et pour chaque paramètre les uns après les autres, nous allons suivre une ligne parallèle à leur axe où des estimations de la performance seront faites à intervalle régulier pour identifier les limites de Ra [Figure 2-15]. Une première approximation de Ra ( $Ra'$ ) assez grossière est ainsi obtenue. Pour l'améliorer, nous considérons le côté le plus important du polyèdre ainsi formé et nous suivons sa médiatrice vers l'extérieur du polyèdre jusqu'à ce que l'on trouve la limite de Ra, définissant ainsi un nouveau polyèdre ( $Ra''$ ). On continue ainsi et on estime que  $Ra'$  approche suffisamment bien Ra lorsque la surface du polyèdre n'évolue plus de manière sensible. Il suffit ensuite de générer aléatoirement N combinaisons de paramètres. Si N' d'entre elles correspondent à un point de l'espace  $Ra'$  que l'on vient de définir (on en connaît ses bornes), le rendement est alors  $N'/N$ .



**Figure 2-15 : Recherche de Ra par approximation polyédrique**

**b) Approximation ellipsoïdale**

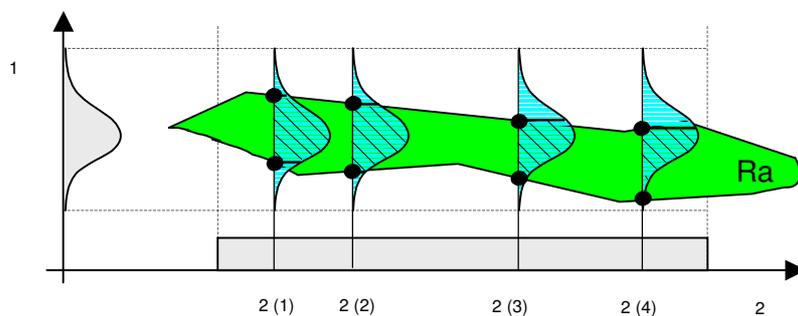
Avec ce type d'approximation,  $R_{a'}$  ne sera plus un polyèdre mais aura une forme d'ellipse [Figure 2-16]. L'utilisation d'une approximation ellipsoïdale par morceau permet ainsi d'utiliser cette méthode malgré une forme éventuellement non convexe de  $R_a$  [KERAa98][BRAH84]. Mais nous restons néanmoins limités par le faible nombre de paramètres traitables.



**Figure 2-16 : Approximation de  $R_a$  par un ellipsoïde et par des fragments d'ellipsoïdes**

**c) Exploration orthogonale**

Cette méthode est aussi appelée ODOS (One Dimensional Orthogonal Search). Elle est utile pour les circuits ayant un comportement linéaire, auquel cas elle permet l'économie de nombreuses simulations. Des valeurs sont d'abord tirées aléatoirement pour un paramètre  $i$ , notées  $i_{(j)}$  [Figure 2-17]. Pour chacun de ces points, on effectue une recherche, parallèlement aux axes de tous les autres paramètres, pour déterminer  $R_a$ . On peut déterminer le rendement en chaque point  $i_{(j)}$  en réalisant l'intégrale de la densité de probabilité des autres paramètres sur l'intervalle où ils correspondent à  $R_a$ . La moyenne de tous ces rendements correspond au rendement global. La variance de l'estimateur du rendement est réduite du fait de la non utilisation de Monte-Carlo pour le calcul des rendements élémentaires, mais d'une simple intégrale [KERAa98] [STYB95].



**Figure 2-17 : Estimation du rendement par exploration orthogonale**

**d) Exploration radiale**

La méthode consiste à générer aléatoirement quelques droites dans  $R_t$  [Figure 2-18]. Sur chacune de ces droites, des points sont positionnés en respectant la densité de

probabilité des paramètres (recherche de la Région d'Acceptabilité). Dans le cas d'une distribution uniforme pour ces paramètres, des résultats expérimentaux ont permis de proposer une formule d'estimation du rendement :

$$\hat{Y}(x) = \frac{1}{2N} \sum_{i=1}^N \{ (Ri^+)^n + (Ri^-)^n \}$$

$$Ri^+ = \begin{cases} \frac{Ra_i^+}{Rt_i^+} & \text{si } Ra_i^+ \leq Rt_i^+ \\ \text{sinon } 1 \end{cases} \quad \text{et} \quad Ri^- = \begin{cases} \frac{Ra_i^-}{Rt_i^-} & \text{si } Ra_i^- \leq Rt_i^- \\ \text{sinon } 1 \end{cases}, \quad N \text{ le nombre de lignes}$$

considérées et n la dimension de l'espace des paramètres de bruit [KERAa98] [DUBO87] [BRAH84].

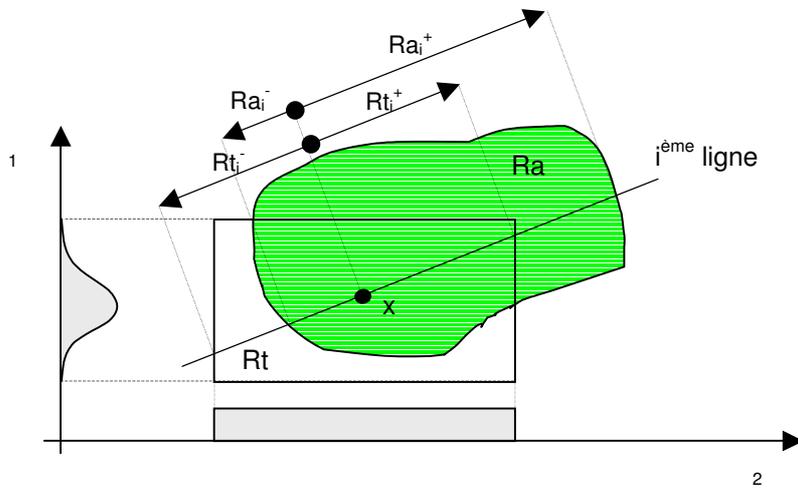


Figure 2-18 : Estimation du rendement par exploration radiale

**e) Intégrale de surface**

Pour cette méthode, on choisira parmi les n paramètres de bruit un paramètre nous évaluerons des rendements locaux le long de lignes parallèles à l'axe de correspondant à un échantillonnage réalisé dans le sous-espace de bruit de dimension n-1 (on passe ainsi d'un "volume" à une "surface", d'où le nom de la méthode) [Figure 2-19].

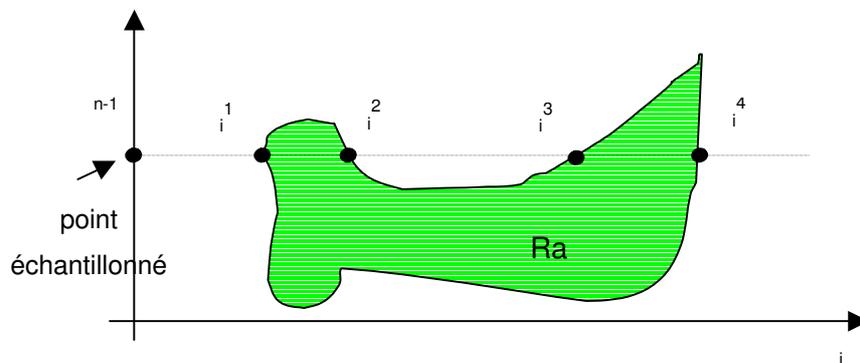


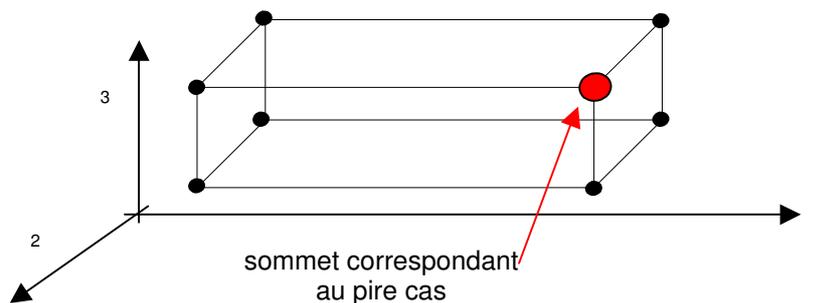
Figure 2-19 : Estimation du rendement par la méthode de l'intégrale de surface

La moyenne de ces rendements est le rendement global. Pour évaluer ces rendements locaux, on utilise une "fonction de densité de probabilité cumulative",  $F_{\theta,i}()$ , que l'on utilise uniquement lorsque l'on rencontre une limite de Ra. Cette fonction est négative lorsqu'en fonction calculées à chaque intersection de la ligne et de Ra pour obtenir le rendement associé à une ligne. Ainsi, le rendement local de la ligne représentée sur la Figure 2-19 s'écrira :  $Y_{\theta_{n-1}} = F_{\theta,i}(\theta_i^2) - F_{\theta,i}(\theta_i^1) + F_{\theta,i}(\theta_i^4) - F_{\theta,i}(\theta_i^3)$  [STYB95] [KERAa98].

### 2.3.4 METHODE DES PIRES CAS

La méthode des pires cas est la base de l'analyse statistique des circuits en microélectronique, mais elle ne permet pas une estimation du rendement [ZEBB96] [KERAa98]. Elle permet par contre de dire si, pour la configuration d'un circuit que l'on estime être la pire en regard d'une performance donnée, la réponse du système sera dans les limites fixées par le concepteur. Si cela est le cas, nous aurons a priori un rendement de 100% vis-à-vis de cette performance. Dans le cas contraire, on ne peut pas estimer le rendement.

Dans les circuits CMOS, on attribue la plus grande part des fluctuations des performances à un petit nombre de paramètres (typiquement quatre). Il y en a souvent beaucoup plus à prendre en considération dans le cas d'un MEMS. Pour utiliser la méthode des pires cas, il est important de décorrélérer ces paramètres, sinon il faut considérer les résultats obtenus comme très pessimistes. On fait pour cela une analyse en composantes principales (p. 77). Prenons un exemple avec trois facteurs de bruit décorrélés [Figure 2-20].



**Figure 2-20 : Sommet correspondant au pire cas dans un espace à trois dimensions**

Pour trouver le pire cas, il faudrait évaluer la performance considérée à chaque sommet de  $R_t$ . A  $n$  paramètres correspond un hyper-parallélépipède de  $2^n$  sommets, rapidement, le nombre de simulations à réaliser devient prohibitif. En pratique, deux méthodes semblent être utilisées plus souvent : *l'arithmétique des intervalles* et *l'analyse des sommets*. Dans ce

dernier cas, on réalise une analyse de sensibilité sur chaque paramètre. On ne se préoccupe donc pas de la forme de sa dispersion, mais on fait l'hypothèse que la performance se comporte comme une fonction monotone vis-à-vis de chaque facteur et on en déduit le sommet qui correspond au pire cas. Il n'y a pas de façon simple de s'assurer de ce comportement monotone, mais la méthode donne généralement des résultats satisfaisants.

## 2.4 OPTIMISATION DU RENDEMENT

Dans cette section, nous faisons un état de l'art des fonctions utilisées comme quantité à optimiser pour améliorer le rendement ainsi que des méthodes utilisées encadrant ces fonctions (souvent appelées fonctions coût) [DIRE92]. Nous détaillerons principalement les méthodes utilisant des plans d'expériences. Leurs particularités nous semblent très intéressantes dans le cas des MEMS.

### 2.4.1 MESURES DU RENDEMENT ET DE LA QUALITE

Lorsque le rendement est directement optimisé, on utilise le terme "direct" pour qualifier les algorithmes ou méthodes utilisés. Les approches utilisant d'autres objectifs en phase d'optimisation afin d'améliorer le rendement sont qualifiées d'indirectes. Nous donnons les fonctions coût les plus couramment utilisées et qui permettent d'améliorer le rendement de manière indirecte.

#### a) Indice de capabilité

Appelé aussi indice d'aptitude, il permet d'évaluer le rendement et la robustesse d'un produit. Considérons une performance telle qu'une fréquence de résonance. Nous désirons que cette dernière soit comprise entre 10 et 11kHz. Si l'on considère un échantillon de plusieurs MEMS, on pourra associer à cette performance une valeur moyenne  $\mu$  et une dispersion  $\sigma$  [Figure 2-21]. Nous considérons toujours qu'en première approximation, les performances comme les paramètres de bruit suivent une distribution de forme gaussienne.

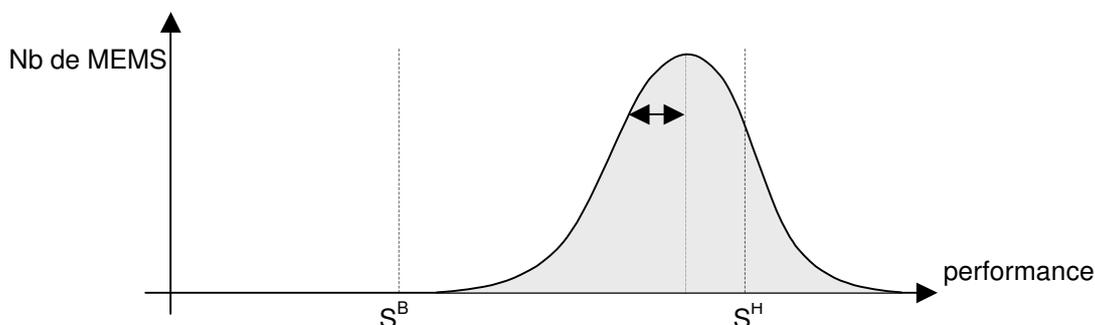


Figure 2-21 : Indice de capabilité d'une performance

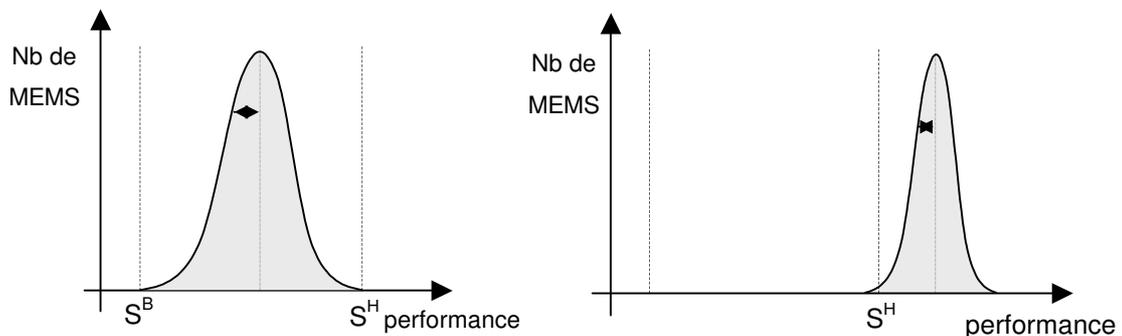
L'indice de capabilité de cette performance est  $C_p = \frac{S^H - S^B}{6\sigma_y}$ , où  $H$  et  $B$  sont utilisés pour

*haut* et *bas*,  $S$  en référence à *spécification*, et  $\sigma_y$  est l'écart type de la performance. Suivant que l'on considère un processus établi ou nouveau, ou que l'on désire s'assurer d'une marge de sécurité, différentes valeurs de  $C_p$  à obtenir ont été fixées. Ces dernières ne sont pas une obligation, elles ont été établies aux vues de nombreuses expériences industrielles [Tableau 2-3].

|  | gabarit | une seule limite |
|--|---------|------------------|
| processus existant                                   | 1,33    | 1,25             |
| nouveau processus                                    | 1,50    | 1,45             |
| paramètre de sécurité ou critique processus existant | 1,50    | 1,45             |
| paramètre de sécurité ou critique                    | 1,67    | 1,60             |

**Tableau 2-3 : Valeurs conseillées de l'indice de capabilité  $C_p$**

La figure [Figure 2-22] représente des performances ayant un indice de capabilité de 1 et de 2. On comprend que plus cet indice est élevé, plus la dispersion est réduite. Si un système a un  $C_p$  inférieur à 1, on dit qu'il n'est pas capable, cela veut aussi dire que le rendement ne peut pas être de 100% ; supérieur à 2, on dit que le système est très performant.



**Figure 2-22 : Indice de capabilité  $C_p = 1$  et  $C_p = 2$**

Toutefois, on ne peut pas se baser uniquement sur ce dernier, car il lui manque une référence à la valeur moyenne de la performance : une performance peut avoir un bon indice de capabilité tout en étant totalement hors spécifications.  $C_p$  permet donc seulement de juger convenables ou pas des performances déjà centrées, c'est à dire dont la valeur moyenne est située au centre d'un gabarit. Dans le cas contraire, on utilise l'indice de capabilité  $C_{pk}$ , défini

par la combinaison des deux indices  $C_{ph} = \frac{S^H - \bar{Y}}{3\sigma_y}$  et  $C_{pb} = \frac{\bar{Y} - S^B}{3\sigma_y}$  par :

$C_{pk} = \min\{C_{ph}, C_{pb}\}$ , où  $\bar{Y}$  est la valeur moyenne de la performance. Tout comme pour  $C_p$ , il existe des valeurs de  $C_{pk}$  pour lesquelles on considère que la performance répond aux attentes, elles sont données dans le Tableau 2-4. Une méthode d'estimation et d'optimisation du rendement basée sur ces indices illustre leur intérêt dans [DEBY98].

| Valeur de Cpk | Evaluation                  |
|---------------|-----------------------------|
| 0,8           | pauvre (incapable)          |
| 1,0           | juste capable               |
| 1,66          | bon                         |
| 2,5           | excellent (très performant) |

**Tableau 2-4 : Valeurs conseillées de l'indice de capabilité  $C_{pk}$**

La réduction de la variabilité d'une performance est beaucoup plus dure à obtenir que l'ajustement de sa valeur moyenne (3.1.2) Nous pouvons définir une fonction  $\phi$ , grâce aux indices  $C_p$  et  $C_{pk}$ , qui nous permettra de mettre l'accent sur l'une ou sur l'autre [KERAa98]

[GIEL98]. Elle s'écrit pour une performance  $i$  :  $\phi_i^b(x) = \frac{S_i^H - S_i^B}{6\sigma_y} + \lambda \frac{S_i^C - \bar{Y}_i}{3\sigma_y}$  et

$\phi_i^h(x) = \frac{S_i^H - S_i^B}{6\sigma_y} + \lambda \frac{\bar{Y}_i - S_i^C}{3\sigma_y}$ , avec  $S_i^C = \frac{S_i^H - S_i^B}{2}$ . On remarque que si  $S_i^C = \bar{Y}$ , le

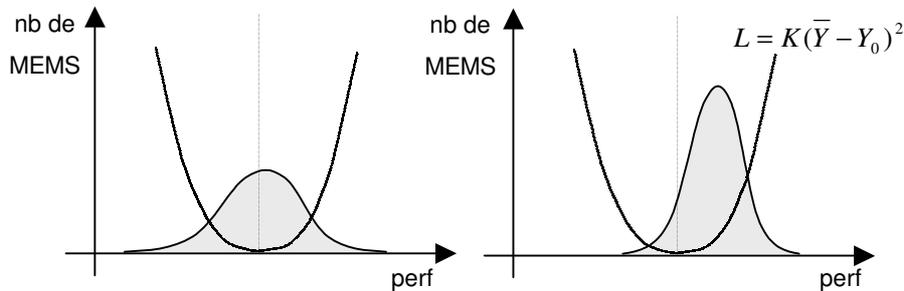
MEMS est alors centré, et on ne travaille que sur l'indice  $C_p$  pour réduire la variabilité. De même pour  $\lambda=0$ . Par contre,  $\lambda=1$  nous renvoie à la formule du  $C_{pk}$ . Donc, plus la réduction de la variabilité aura d'importance dans le problème d'optimisation, plus la valeur de  $\lambda$  sera proche de 0. Par ailleurs, cette dernière peut évoluer au cours de l'optimisation. Ce dernier s'écrira :  $\min_x \left\{ \min_i \left\{ \phi_i^h(x), \phi_i^b(x) \right\} \right\}$ .

Certains logiciels, tel Aspire [ASPI-net], utilisent l'indice  $C_{pk}$  pour améliorer le rendement.

### b) Fonction Perte de Taguchi

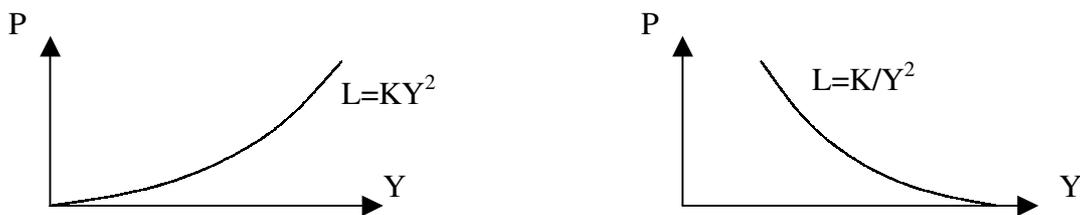
Il s'agit d'une autre fonction coût à minimiser pour améliorer le rendement. Définie par G. Taguchi qui l'a nommée *fonction Perte* [PILL01] ( $L = K(\bar{Y} - Y_0)^2$  ou EQL pour *Expected Quality Loss*). Elle permet elle aussi d'ajuster la valeur moyenne tout en réduisant la

variabilité. On peut démontrer [PILL01] que  $\bar{L} = \frac{1}{n} \sum K(Y_i - Y_0)^2 = K(\sigma^2 + (\bar{Y} - Y_0)^2)$  dans le cas de la recherche d'une cible  *nominale*. Plusieurs solutions peuvent minimiser cette fonction, suivant que l'on met l'accent sur la réduction  $\sigma^2$  ou de  $(\bar{Y} - Y_0)^2$  [Figure 2-23].



**Figure 2-23 : Deux fonctions Perte de même valeur moyenne dans le cas d'un échantillon**

Mais cette fonction Perte sera différente dans le cas de la recherche d'un maximum (par exemple un déplacement) ou d'un minimum (par exemple une déflexion) [Figure 2-24].



**Figure 2-24 : Fonction Perte dans le cas où l'optimum est un minimum ou un maximum**

Enfin, Taguchi a défini un rapport signal/bruit que l'on essaie de maximiser, ce qui se traduit par une minimisation de la fonction Perte. L'intérêt du rapport est d'utiliser le logarithme qui tend à linéariser la réponse, un facteur 10 pour utiliser des décibels, et le signe approprié pour chercher logiquement à maximiser le rapport signal/bruit [Tableau 2-5]. Taguchi a proposé de nombreuses autres fonctions Perte dédiées à des cas plus particuliers. Le choix de la fonction coût est critique pour un processus d'optimisation, et s'effectue en fonction de la nature du problème.

| type de cible | perte par produit        | perte moyenne  | rapport signal/bruit   |
|---------------|--------------------------|--|--|
| nominale      | $L = K(\bar{Y} - Y_0)^2$ | $\bar{L} = K(\sigma^2 + (\bar{Y} - Y_0)^2)$                        | $-10 \log (\sigma^2 + (\bar{Y} - Y_0)^2)$  |
| minimale      | $L = KY^2$               | $\bar{L} = K(\sigma^2 + \bar{Y}^2)$                                | $-10 \log (\sigma^2 + \bar{Y}^2)$  |
| maximale      | $L = K / Y^2$            | $\bar{L} = \frac{K}{\bar{Y}^2} (1 + 3 \frac{\sigma^2}{\bar{Y}^2})$ | $-10 \log \left( \frac{1}{\bar{Y}^2} (1 + 3 \frac{\sigma^2}{\bar{Y}^2}) \right)$ |

**Tableau 2-5 : Principaux cas et rapports signal/bruit associés (Taguchi)**

Il semble donc que la réduction de la variabilité et l'ajustement de la valeur moyenne soient traités au même niveau. C'est ce qui fait le danger de cette fonction, car en pratique, il est beaucoup plus aisé de modifier la valeur moyenne d'une performance que d'en réduire la variabilité. Cette méthode est donc intéressante si l'on réalise plusieurs simulations et que l'on désire choisir, en étudiant leurs résultats, quelles sont les meilleures valeurs à donner aux paramètres de dessin (voir les plans produits du §2.4.3). Mais il est donc à craindre, dans le cas d'un algorithme qui commanderait de nouvelles simulations après traitement des résultats des précédentes, que l'ajustement de la valeur moyenne se fasse beaucoup plus rapidement que la réduction de variabilité, souvent très faible.

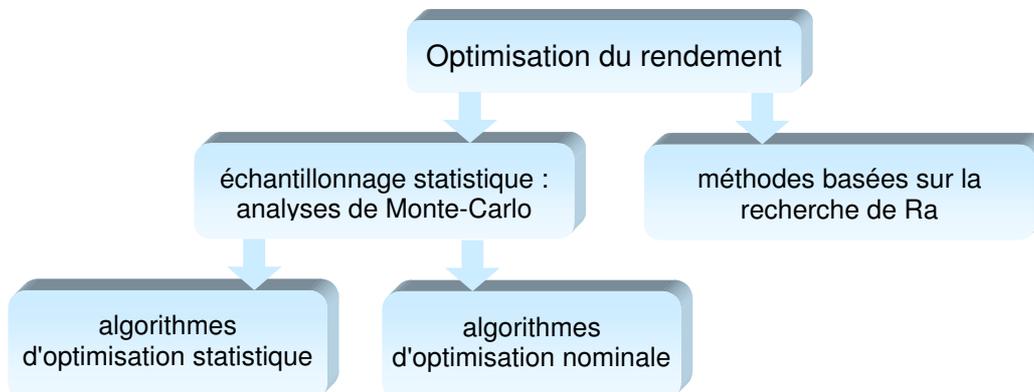
### c) Autres fonctions coût

Pour éviter l'effet pervers d'une double optimisation qui se ferait au détriment de la réduction de la variance, on peut se contenter de minimiser la variance dans un premier temps, puis dans un deuxième temps, d'ajuster la valeur moyenne. Si l'on part d'une bonne solution initiale (cela est encore plus recommandé dans le cas de l'optimisation statistique que de l'optimisation nominale), on peut prendre garde, lors de la réduction de variance, à ne pas utiliser les paramètres de dessin qui nous éloigneraient trop de la valeur moyenne initiale. On peut toutefois y revenir pendant la deuxième étape en évitant d'utiliser les facteurs qui ont une grande influence sur la variance, pour ne pas perdre le gain précédent. Cette approche nécessite donc la définition de fonctions coût différentes de celle de Taguchi où l'erreur et la variabilité de la réponse ne sont pas traitées simultanément.

## 2.4.2 METHODES D'OPTIMISATION DU RENDEMENT

On peut les classer comme on l'a fait pour les méthodes d'estimation du rendement [Figure 2-25]. Les approches de type géométrique (recherche de Ra) sont le plus souvent

indirectes. Dans la littérature, il y a peu de méthode d'optimisation du rendement qui soit déterministe sans avoir recours à une forme d'échantillonnage [SCHW99].



**Figure 2-25 : Méthodes d'optimisation du rendement**

### **a) Algorithmes d'optimisation nominale**

Les algorithmes que l'on peut rencontrer dans le cadre de l'optimisation nominale sont utilisables pour l'optimisation statistique. La fonction à optimiser la plus basique est alors le rendement. Le problème est que l'estimation de cette fonction à chaque nouvelle itération de l'algorithme demande des dizaines de milliers de simulations. On peut essayer d'améliorer l'efficacité de l'algorithme en diminuant le nombre de ces simulations, mais cela augmente la variance de l'estimateur de Monte-Carlo. Le bruit statistique ainsi généré dégrade l'efficacité de l'algorithme qui peut ne plus converger même si une solution "simple" existe. Une autre solution est d'utiliser des plans d'expériences (§2.4.3) pour construire à peu de frais (nombre minimum de simulations) un polynôme modélisant la performance ciblée. Mais si le nombre de paramètres est important, la construction d'un tel modèle peut s'avérer très onéreuse et donc finalement peu intéressante. On peut toutefois retrouver certains de ces algorithmes dans la littérature sur l'optimisation statistique [MA01] [PAN93] [SCHE01].

Les méthodes utilisant le gradient du rendement sont nombreuses. On peut le traiter suite à une analyse de Monte-Carlo, ainsi qu'éventuellement les dérivées secondes par rapport aux paramètres et à leurs tolérances. On en déduit les valeurs appropriées des paramètres pour la prochaine itération. L'utilisation de l'échantillonnage paramétrique (§2.4.2) permet ici de faire l'économie de nombreuses simulations.

Les algorithmes les plus fréquemment cités dans le cadre de notre étude sont : l'approche du simplexe, l'utilisation des sensibilités marginales [DUBO87] [BRAH84], l'approximation stochastique [KERAA98] [STYB95], le recuit simulé [ZEBB96], l'exploration axiale... L'approche de l'intégrale de surface [p. 57] peut aussi être utilisée. Elle permet de définir le gradient du rendement et de travailler dessus. Mais on retrouve donc ici les

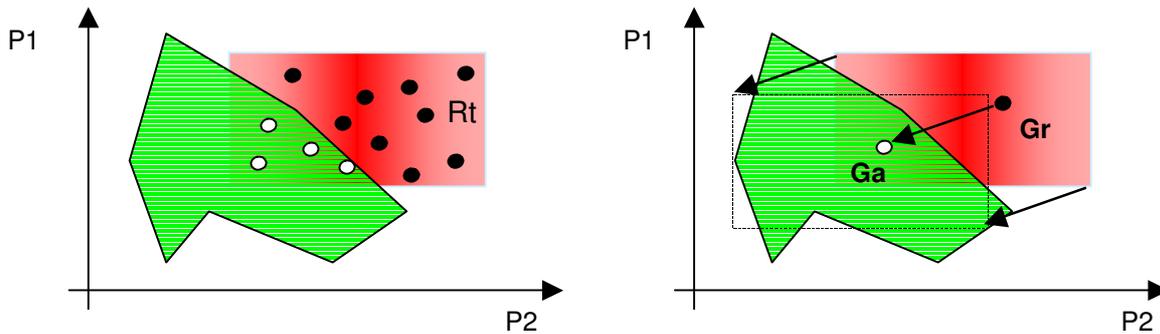
problèmes énoncés lors de l'utilisation d'algorithmes initialement prévus pour l'optimisation nominale : le nombre de simulations est prohibitif et nécessite l'emploi d'une fonction de performance, qui peut elle même être délicate à générer.

### b) Algorithmes d'optimisation statistique

Nous donnons ici un aperçu des principaux algorithmes, mais ils sont souvent associés à des approches différentes qui donnent autant de variantes en pratique [STYB90].

#### b.1 Centre de gravité (barycentre)

Il s'agit de la méthode qui semble être la plus simple et robuste utilisée dans ce domaine, elle est d'ailleurs citée par toutes les thèses référencées au chapitre précédent. Elle est notamment implémentée dans Aspire [ASPI-net]. Un échantillonnage est réalisé dans  $R_t$  [Figure 2-26]. Les points noirs sont hors spécifications et entraînent une baisse du rendement. Les points blancs sont dans les spécifications : ils appartiennent à  $R_t$  ET  $R_a$ .



**Figure 2-26 : Optimisation du rendement par l'algorithme du centre de gravité**

Soit  $N=N_a+N_r$  le nombre total de simulations. On calcule le barycentre  $G_a = [G_{a_1}, \dots, G_{a_k}]^T$  des  $N_a$  points acceptables avec  $G_{a_i} = \frac{1}{N_a} \sum_{j=1}^{N_a} P_i^j$  pour  $i=1, \dots, k$ . De même, le barycentre des  $N_r$  points rejetés est donné par  $G_{r_i} = \frac{1}{N_r} \sum_{j=1}^{N_r} P_i^j$ . Le point  $p^{m+1}$  à l'itération  $m+1$  peut s'exprimer en fonction du point  $p^m$  établi à l'itération  $m$  par  $p^{m+1} = p^m + \lambda(G_a^m - G_r^m)$ . est appelé "facteur de progression". C'est une variable assimilable à un pas qui peut être dynamique. C'est une valeur qui oscille généralement entre 0,5 et 1,5. On peut aussi imaginer des variantes telles que :

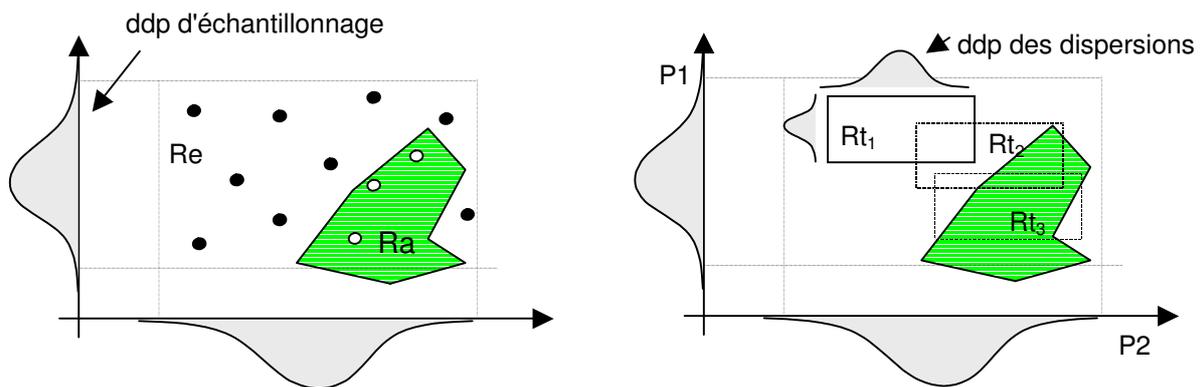
$$\text{-rendement), } p^{m+1} = p^m + \lambda(G_a^m - p^m), \quad p^{m+1} = G_a^m \dots$$

Encore une fois, le nombre de simulations nécessaires diminue énormément l'efficacité de cet algorithme au cours des itérations. Pour tenter d'y remédier, on peut l'utiliser avec la

méthode des *points communs*. Lorsque  $R_t$  varie peu d'une itération à l'autre, on peut conserver bon nombre de résultats de l'analyse de Monte-Carlo précédente : tous ceux qui appartiennent à la fois à  $R_t^m$  et  $R_t^{m+1}$ . L'analyse de Monte-Carlo de l'itération  $p^{m+1}$  s'en trouvera allégée d'autant. Il faudra bien sûr prendre garde à redéfinir les bornes dans lesquelles les paramètres peuvent varier au cours de l'analyse. Malheureusement, l'utilisation de cette méthode devient complexe lorsque les distributions des paramètres ne sont pas uniformes, ce qui est généralement le cas.

### b.2 Echantillonnage paramétrique

Cette méthode, basée sur l'échantillonnage pondéré [p 54], a été développée initialement pour les circuits discrets. On définit une Région d'Exploration  $Re$  dans l'espace des paramètres, dont on estime qu'elle englobe  $R_t$  initiale et une bonne partie de  $Ra$  [KERAa98] [STYB95].

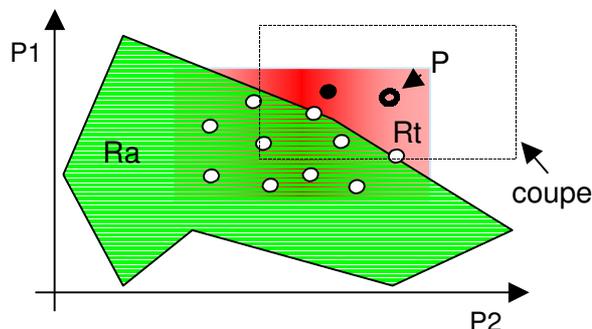


**Figure 2-27 : Optimisation du rendement par l'échantillonnage paramétrique**

On réalise un grand nombre de simulations, par exemple 1000, en échantillonnant l'ensemble des paramètres et en suivant une loi de distribution normale. Les résultats de cette analyse de Monte-Carlo sont rangés dans une base de données. On peut de la sorte distinguer une partie importante de  $Ra$ , calculer le gradient du rendement pour différentes  $R_t$  et en déduire des valeurs pour les paramètres susceptibles d'améliorer le rendement. Cette méthode donne de meilleurs résultats lorsque les densités de probabilités (ddp) utilisées lors de l'échantillonnage dans la région d'exploration sont proches des ddp associées à chaque paramètre. Mais au cours des itérations,  $R_t$  qui évolue peut se retrouver sur le bord de  $Re$ , il peut être alors intéressant de redéfinir  $Re$  centrée sur  $R_t$  ( $R_{t3}$  dans l'exemple). On peut utiliser la méthode des points communs pour économiser des simulations lors de cette étape.

### b.3 Méthode des coupes

Cette méthode consiste à effectuer un échantillonnage de  $R_t$ , centrée en  $P$  [Figure 2-28].



**Figure 2-28 : Réalisation d'une coupe pour un point violant les spécifications**

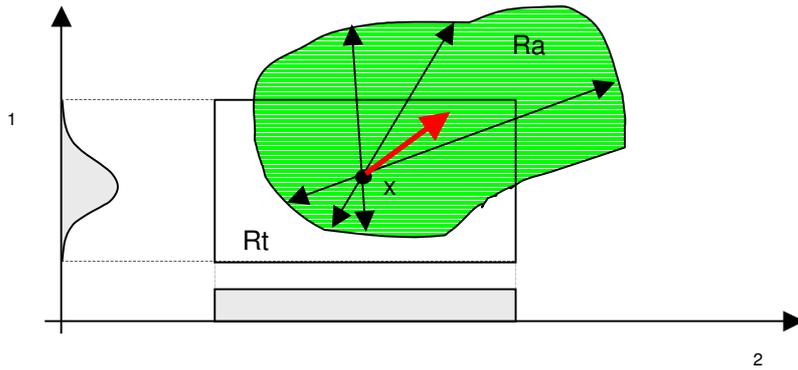
Si l'un des points échantillonné n'appartient pas à  $R_a$ , on arrête l'analyse de Monte-Carlo et on déclare que la zone (de même dimension que  $R_t$ ) autour de ce point est inintéressante. On réalise ainsi une coupe. Lors de la prochaine itération, un nouveau point  $P$  devra être choisi. Ce dernier ne devra pas appartenir à l'union des coupes réalisées auparavant, car le rendement ne pourrait y être de 100%. Le choix du nouveau point  $P$  peut se faire en partie à l'aide de l'algorithme du centre de gravité, ou autre. Telle quelle, cette méthode ne fonctionnera que s'il existe un point pour lequel le rendement est de 100%, ce qui n'est a priori pas évident [ZEBB96].

### c) Méthodes d'optimisation géométriques

Les méthodes géométriques les plus courantes sont basées sur des méthodes géométriques d'estimation du rendement que nous avons vues au chapitre précédent.

#### c.1 Exploration radiale

On reprend ici la méthode d'exploration radiale développée pour l'estimation du rendement [p. 56]. Dans cette approche géométrique, les limites de  $R_a$  étaient recherchées et localisées par les vecteur  $R_a^+$  et  $R_a^-$  pour différentes lignes [Figure 2-29].

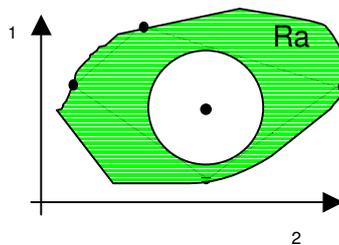


**Figure 2-29 : Centrage d'un circuit par la méthode de l'exploration radiale**

La somme des vecteurs  $Ra^+ + Ra^-$  donne la direction dans laquelle déplacer  $Rt$  pour augmenter le recouvrement avec  $Ra$ , et donc le rendement. Il faut pondérer l'amplitude de ce déplacement en fonction du nombre de lignes lui ayant apporté une contribution notable. Cette méthode est heuristique, mais donne de bons résultats, notamment pour les circuits linéaires. Encore une fois, la méthode des points communs peut être utilisée.

### c.2 Approximation polyédrique

Cette méthode est aussi basée sur une méthode d'estimation du rendement [p. 55]. Le principe est de trouver le centre de l'hypersphère inscrite dans le polyèdre définissant  $Ra$ . Le nombre de simulations augmentant exponentiellement avec le nombre de paramètres, cette méthode se limite aux cas n'en traitant pas plus de cinq.



**Figure 2-30 : Hypersphère inscrite dans le polyèdre définissant  $Ra$**

La méthode de distance pire cas [SCHW99], développée pour des circuits discrets, consiste à maximiser le rayon de l'hypersphère.

Nous avons passé en revue différentes approches utilisées dans le cadre de l'optimisation du rendement paramétrique. La majorité d'entre elles, développées initialement pour les circuits discrets, ont pu être étendues aux circuits intégrés et donc aux MEMS. La plupart des méthodes rencontrées dans la littérature sont aussi condamnées à ne pouvoir traiter que les cas où peu de paramètres sont considérés dans le processus d'optimisation.

Cela peut évoluer avec les améliorations qui leur sont apportées au fil du temps, mais aussi avec l'augmentation de la puissance des ordinateurs et de l'efficacité des simulateurs.

### 2.4.3 PLANS D'EXPERIENCES

Un plan d'expérience (*Design Of Experiments* ou DOE) est une stratégie optimale permettant de prédire avec le maximum de précision une réponse à partir d'un nombre minimal d'essais et en utilisant un modèle postulé [OZIL97]. Les résultats de ces simulations peuvent être exploités de différentes manières pour en extraire des informations sur les influences des paramètres du problème. Parmi elles, la méthode des surfaces de réponses permet de construire un polynôme modélisant une performance ciblée du MEMS. Ce polynôme peut être utilisé dans le cadre de l'estimation ou de l'optimisation du rendement. Un des avantages de la méthode est qu'il n'y a pas d'hypothèse faite sur la forme de la région d'Acceptabilité, des distributions des paramètres ou des performances.

#### a) Notion de qualité

Les plans d'expériences apparaissent en 1925 avec les travaux de Fisher. Les premières applications ont été pour l'agriculture : on comprend ici l'importance de réduire le nombre d'expériences (simulations) pour connaître l'influence de différents paramètres. Le Docteur G. Taguchi, dans les années 70, a su rendre les plans d'expériences plus lisibles et exploitables pour l'industrie, où l'on peut les retrouver aujourd'hui à tous les niveaux.

Les plans d'expériences sont un outil faisant partie de l'arsenal utilisé pour la maîtrise de la qualité. Cette dernière est devenue critique pour la compétitivité des entreprises à l'heure de la mondialisation. Elle peut s'appliquer à tous les secteurs, tous les processus identifiés et les produits d'une entreprise. Dans le cadre de la production, il est important de s'occuper de qualité dès la phase de conception du produit, c'est à ce niveau que l'on obtient le plus gros retour sur investissement : il est bien moins cher de réévaluer les dimensions d'un MEMS pour le rendre plus robuste que de se trouver face à des clients mécontents des performances du produit. Le remplacement et d'éventuelles indemnités sont facilement estimables, mais l'impact sur l'image de marque est tout aussi important bien que moins facile à évaluer.

#### b) Notions sur les plans d'expériences

On se sert la plupart du temps des plans d'expériences pour construire un polynôme qui modélise la performance ciblée d'un système [DEAL97]. Ses termes seront les paramètres ou  $(x_i)$  du problème, et ses coefficients refléteront leur importance spécifique sur la performance ciblée. On choisira le plan en fonction du polynôme qui correspond le mieux

au processus que l'on souhaite modéliser. On peut aussi exploiter un plan pour définir seulement des tendances sans se préoccuper de la construction d'un polynôme [DEVA94]. Dans cette discipline, les paramètres sont nommés "facteurs" de dessin ou de bruit.

### b.1 Différents modèles

#### **Modèle linéaire sans interactions :**

$y(x) = b_0 + \sum_{i=1}^n a_i x_i$ , ce modèle nécessite  $n+1$  simulations pour être défini. Il est souvent

utilisé pour faire du "factor screening", c'est à dire identifier par une approche de type analyse de sensibilité les facteurs non critiques pour éventuellement les éliminer du processus d'optimisation qu'ils alourdissent inutilement. L'efficacité de cette méthode et de certaines de ses variantes est discutée dans [TROC00], [MOOR00] et [NAZZ00].

#### **Modèle linéaire avec toutes les interactions :**

$y(x) = b_0 + \sum_{i=1}^n a_i x_i + \sum c_{ij} x_i x_j + \sum h_{ijk} x_i x_j x_k + \dots + \sum p_{ijk\dots n} x_i x_j x_k \dots x_n$ , ce modèle nécessite

au moins  $2^n$  simulations pour être défini. Bien sûr, si l'on estime que l'un des termes n'est pas intéressant, on ne le fera pas apparaître dans le polynôme et l'on s'économisera ainsi une simulation. C'est généralement le cas des interactions d'ordre supérieur à 2 qui ont très rarement un rôle significatif à jouer d'un point de vue physique.

#### **Modèle linéaire avec interactions d'ordre 1 (modèle factoriel) :**

$y(x) = b_0 + \sum_{i=1}^n a_i x_i + \sum c_{ij} x_i x_j$ , (équ. 2-3) ce modèle nécessite au moins

$n+1+n(n-1)/2$  simulations pour fixer ses coefficients. On démontrera plus loin que c'est le mieux adapté dans le cadre d'optimisation du rendement pour les MEMS. Il est utilisé notamment lorsque l'on cherche la direction d'un optimum.

#### **Modèle du second degré :**

$y(x) = b_0 + \sum_{i=1}^n a_i x_i + \sum c_{ij} x_i x_j + \sum e_{ii} x_i^2$ , c'est le modèle factoriel auquel on a ajouté les

termes quadratiques, il nécessite donc au moins  $2n+1+n(n-1)/2$  simulations pour être complètement défini. Il est utilisé lorsque l'on veut localiser l'optimum : dans ce cas, on fait l'hypothèse qu'il se trouve dans le domaine de validité du polynôme.

b.2 Exemple

Les plans d'expériences se présentent sous forme de matrices. Prenons l'exemple d'un plan à 2 facteurs  $X_1$  et  $X_2$  auxquels on associe 2 niveaux [Tableau 2-6]. On estime qu'un modèle factoriel modélisera bien la réponse de notre système, par exemple la déflexion à l'extrémité d'une poutre :  $y(x) = b_0 + a_1x_1 + a_2x_2 + c_{12}x_1x_2$ . On peut voir sur le Tableau 2-7 l'allure possible d'un plan permettant de définir ce modèle. Il possède 4 coefficients et nécessite donc au moins 4 expériences (simulations) pour être construit.

|                      | Largeur d'une poutre ( $X_1$ ) | Longueur de la poutre ( $X_2$ ) |
|----------------------|--------------------------------|---------------------------------|
| Niveau bas (- ou 1)  |                                |                                 |
| Niveau haut (+ ou 2) |                                |                                 |

**Tableau 2-6 : Valeurs plausibles pour les facteurs : niveaux**

Le modèle choisi ne sera valable que dans l'intervalle fixé par les niveaux haut et bas pour chaque facteur. L'écart entre les 2 niveaux est important : trop faible, l'effet attribué à ce facteur ne sera pas significatif. Trop important, un modèle simple (factoriel) risque de ne pas être fiable pour cet intervalle.

|                | $X_1$ | $X_2$ | $X_1 * X_2$ | Y       |
|----------------|-------|-------|-------------|---------|
| simulation n°1 | -     | -     | +           | $Y_1=2$ |
| simulation n°2 | -     | +     | -           | $Y_2=4$ |
| simulation n°3 | +     | -     | -           | $Y_3=1$ |
| simulation n°4 | +     | +     | +           | $Y_4=5$ |

**Tableau 2-7 : Matrice d'un plan d'expérience pour 2 facteurs à 2 niveaux**

Chaque ligne de ce plan correspond à une simulation à réaliser pour laquelle les valeurs des facteurs nous sont imposées, la dernière colonne étant dédiée aux résultats des expériences (ou simulations). On associe une colonne à chaque paramètre. Ici, la troisième colonne correspond à une interaction : c'est le produit des colonnes 1 et 2 qui nous aidera à calculer le coefficient  $c_{12}$  associé à l'interaction. Pour calculer les coefficients, appelés aussi "effets moyens", on fait la moyenne des réponses de tous les essais réalisés à un niveau donné. Pour l'effet de A au niveau i :

$$E_{Ai} = \text{Moyenne des réponses lorsque A est au niveau i} - \text{Moyenne générale}$$

Ce plan à 2 niveaux et 2 facteurs est un plan factoriel complet : les 2<sup>2</sup> combinaisons possibles sont étudiées. Dans le cas de facteurs à 2 niveaux, on vérifie  $E_{x_1+} + E_{x_1-} = 0$ . Les calculs donnent :

$$M = \frac{Y_1 + Y_2 + Y_3 + Y_4}{4} = \frac{2 + 4 + 1 + 5}{4} = 3$$

$$E_{x_{1+}} = \frac{Y_3 + Y_4}{2} - M = \frac{1 + 5}{2} - 2.5 = -0,5$$

$$E_{x_{2+}} = \frac{Y_2 + Y_4}{2} - M = \frac{4 + 5}{2} - 2.5 = 2$$

Lorsque  $X_1$  est au niveau bas (-), l'effet de  $X_2$  sur la déflexion est :  $Y_2 - Y_1 = 4 - 2 = 2$

Lorsque  $X_1$  est au niveau haut (+), l'effet de  $X_2$  sur la déflexion est :  $Y_4 - Y_3 = 5 - 1 = 4$

Comme l'effet de  $X_1$  sur  $Y$  est fonction de  $X_2$ , on en déduit qu'il existe une interaction entre ces 2 facteurs. L'effet de l'interaction entre  $X_1$  et  $X_2$  quand ils sont aux niveaux  $i$  et  $j$  s'écrit :

$$I_{x_1 i x_2 j} = (\text{Moyenne des réponses lorsque } X_1 \text{ au niveau } i \text{ et } X_2 \text{ au niveau } j) - M - E_{x_1 i} - E_{x_2 j}$$

Dans ce type de plan, on calcule un type d'interaction, les 3 autres en découlent avec

$$I_{x_1+x_2+} = I_{x_1-x_2-} = I_{x_1-x_2+} = I_{x_1+x_2-}$$

soit dans notre cas :

$$I_{x_1+x_2+} = \frac{Y_4}{1} - M - E_{x_1+} - E_{x_2+} = 5 - 3 - (-0,5) - 2 = 0,5$$

Nous avons défini tous les termes nécessaires pour la construction d'un modèle matriciel de la réponse  $Y_{(x_1, x_2)}$  du MEMS qui s'écrira donc :

$$Y = M + \begin{bmatrix} E_{x_1-} & E_{x_1+} \end{bmatrix} [x_1] + \begin{bmatrix} E_{x_2-} & E_{x_2+} \end{bmatrix} [x_2] + {}^T [x_1] \begin{bmatrix} I_{x_1-x_2-} & I_{x_1-x_2+} \\ I_{x_1+x_2-} & I_{x_1+x_2+} \end{bmatrix} [x_2]$$

où  $[x_i] = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$  au niveau bas,  $[x_i] = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$  au niveau haut, et  ${}^T [x_1]$  la transposée de  $[x_1]$ .

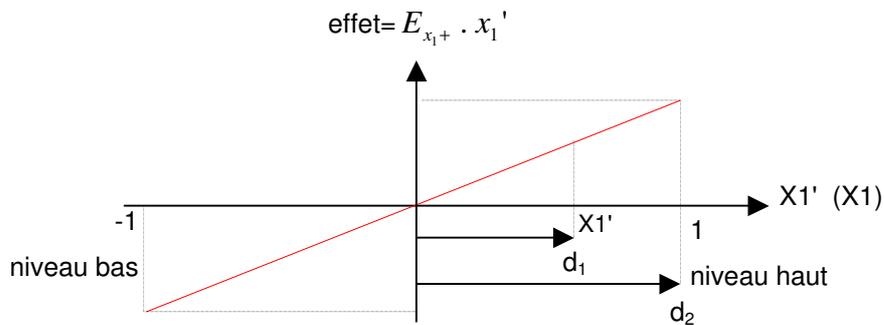
L'application numérique donne :

$Y = 3 + [0,5 \quad -0,5][x_1] + [-2 \quad 2][x_2] + [x_1] \begin{bmatrix} 0,5 & -0,5 \\ -0,5 & 0,5 \end{bmatrix} [x_2]$ . On peut vérifier avec la

simulation n°1 :

$$Y_1 = 3 + [0,5 \quad -0,5] \begin{bmatrix} 1 \\ 0 \end{bmatrix} + [-2 \quad 2] \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} 0,5 & -0,5 \\ -0,5 & 0,5 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = 3 + 0,5 - 2 + 0,5 = 2$$

Mais ce qui est intéressant, c'est d'estimer Y pour n'importe quelle valeur des facteurs dans l'intervalle de validité des niveaux sans pour cela avoir à réaliser une nouvelle expérience. On va pour cela transformer le modèle matriciel en polynôme. Dans notre cas, on l'obtient immédiatement avec :  $Y = M + E_{x_1+} x_1' + E_{x_2+} x_2' + I_{x_1+x_2} x_1' x_2'$  ou  $x_i'$  est un facteur centré qui vaut 1 lorsque  $x_i$  est maximum et  $-1$  lorsque  $x_i$  est au minimum [Figure 2-31].



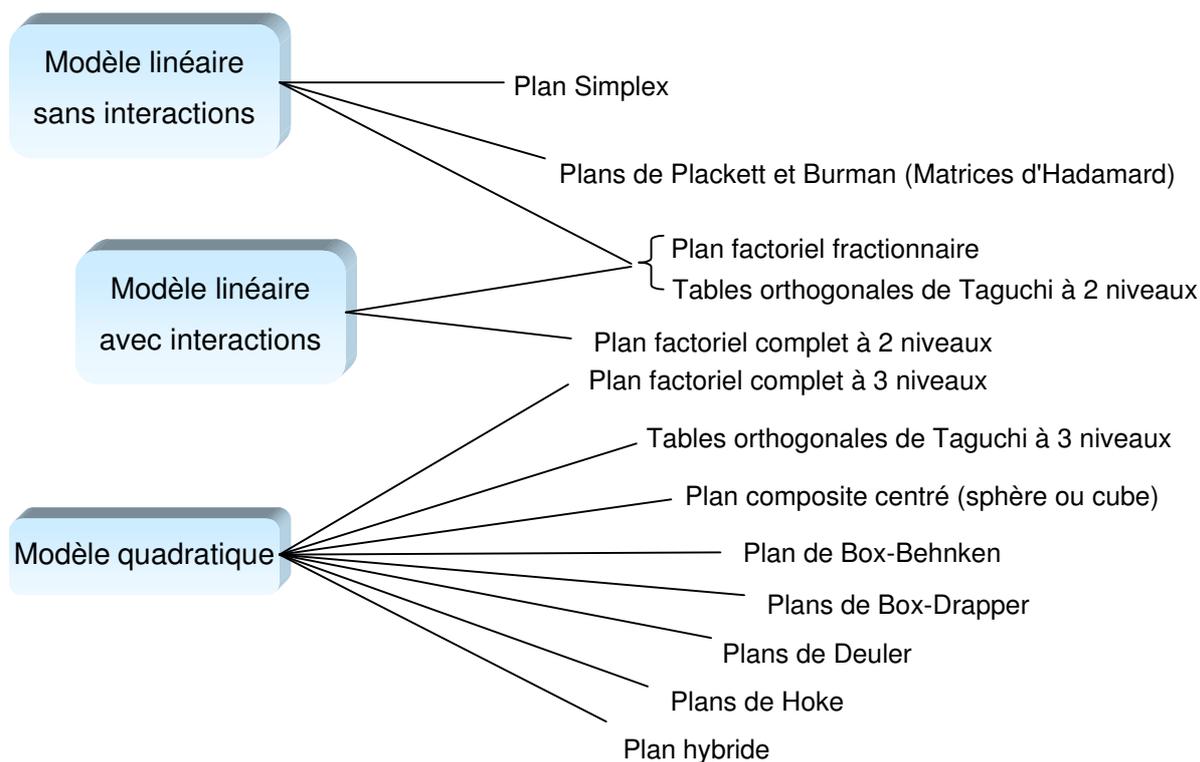
**Figure 2-31 : Utilisation de facteurs centrés pour construire un polynôme**

Le théorème de Thalès donne :  $\frac{E_{x_1+} \cdot x_1'}{d_1} = \frac{E_{x_1+} \cdot 1}{d_2}$  donc  $x_1' = \frac{d_1}{d_2} = \frac{x_1 - \frac{x_{1-} + x_{1+}}{2}}{\frac{x_{1+} - x_{1-}}{2}}$  et le

polynôme s'écrit alors  $Y = 3 - 0,5 x_1' + 2 x_2' + 0,5 x_1' x_2'$ . On peut vérifier avec  $Y_2$  où  $X_1$  est au minimum et  $X_2$  au maximum :  $Y_2 = 3 - 0,5 (-1) + 2 (1) + 0,5 (-1)(1) = 5$ . Pour vérifier la bonne tenue du modèle, on réalise souvent une expérience supplémentaire en prenant des valeurs moyennes pour les facteurs et en comparant ensuite avec les résultats fournis par le polynôme avec les mêmes valeurs.

### b.3 Différents plans d'expériences

On choisit un polynôme et un plan d'expérience en fonction du problème à résoudre. Le plan précédent pris en exemple est une table orthogonale de Taguchi, mais il en existe de nombreux autres [Figure 2-32] [GE99] [OZIL97].



**Figure 2-32 : Plans d'expériences usuels**

Dans le cas précédent, nous ne considérons que 2 facteurs à 2 niveaux. Un problème comportant 15 facteurs nécessiterait  $2^{15} = 32768$  expériences, ce qui est prohibitif, surtout si on projette d'utiliser un plan de manière itérative. Ces plans, dits complets, permettent de définir, outre les effets principaux de chaque facteur, toutes les interactions possibles entre 2 ou plusieurs facteurs, d'où le nombre important d'expériences requises (autant d'expériences que d'inconnues). Or, les interactions du troisième ordre ( $X_1X_2X_3\dots$ ) ou plus sont généralement physiquement négligeables. De plus, toutes les interactions d'ordre 2 ( $X_2X_3\dots$ ) ne figurent pas forcément dans le modèle choisi pour la réponse ciblée. Il en résulte que réaliser un plan complet est excessif : un plan optimal définirait autant d'expériences que d'inconnues dans le modèle. Pour cela, on utilise les plans factoriels fractionnaires, dont l'idée est d'utiliser les fractions utiles de plans complets.

#### b.4 Matrices d'Hadamard

Les matrices d'Hadamard sont des matrices  $X$  vérifiant :  ${}^T X \cdot X = (k + 1)I_{k+1}$  avec  $k$  nombre de facteurs,  $I$  matrice unité. Il s'agit donc de matrices carrées vérifiant  ${}^T H_n \cdot H_n = nI_n$ , dont les éléments valent +/-1 et qui n'existent que pour  $n=1,2$  ou  $n$  multiple de 4. On peut les construire rapidement par la méthode suivante :

$H_{2^n} = \begin{pmatrix} H_n & H_n \\ H_n & H_{-n} \end{pmatrix}$  à partir de  $H_1 = (1)$ , utilisée pour  $2^n - 1$  facteurs,  $n$  entier. On calcule donc

facilement  $H_2, H_4, H_8, H_{16}, H_{32}, H_{64} \dots$ . L'inconvénient de cette méthode est évident : pour un cas à 33 facteurs par exemple, il faudrait utiliser  $H_{64}$ , ce qui revient à réaliser 30 expériences superflues. Cette méthode n'est donc adaptée que pour un nombre de facteurs légèrement inférieur à  $2^n - 1$ . On peut utiliser la méthode de Plackett et Burman pour construire des matrices d'Hadamard pour  $4n - 1$  facteurs, ce qui limite énormément le nombre d'expériences inutiles. Plackett et Burman [PLAC46] fournissent la 1<sup>ère</sup> ligne de la matrice à construire, les autres sont déduites de celle-là par permutation circulaire, la dernière ligne ne contient que des -. Par exemple, pour construire  $H_8$  ( $k=7$  facteurs), la première ligne est ++++--, la

matrice est donc :  $H_8 = \begin{pmatrix} + & + & + & - & + & - & - \\ - & + & + & + & - & + & - \\ - & - & + & + & + & - & + \\ + & - & - & + & + & + & - \\ - & + & - & - & + & + & + \\ - & - & + & - & - & + & + \end{pmatrix}$

Mais ces "1<sup>ères</sup> lignes" ne sont pas données dans la littérature pour des matrices d'un ordre élevé. Il n'existe pas de méthodes permettant de construire une matrice d'Hadamard d'un ordre quelconque : ces dernières sont parfois référencées après avoir été proposées et que leurs caractéristiques aient été vérifiées.

#### b.5 Notion de résolution

Les facteurs ( $P_1$ ) sont dits actions d'ordre 1. Les interactions du second ordre ( $P_1P_2$ ), actions d'ordre 2, etc. On a vu que lors d'un plan complet, les coefficients du polynôme modélisant la réponse étaient définissables pour toutes les actions possibles. Ce n'est pas le cas dans les plans fractionnaires, où l'estimation de ces coefficients peut être faussée par l'existence d'une interaction que l'on aurait décidé de négliger au niveau de la modélisation. Dans les plans de résolution III, on risque de confondre les effets d'une action d'ordre I avec ceux d'une action d'ordre II. On dit dans ce cas que ces actions sont aliasées. On définit ainsi plusieurs résolutions :

résolution III : les actions d'ordre I et II ont un risque d'alias (par exemple,  $P_1$  et  $P_2P_3$ ).

résolution IV : les actions d'ordre I et III, ou II et II, ont un risque d'alias (par exemple,  $P_1$  et  $P_2P_3P_4$  ou  $P_1P_2$  et  $P_3P_4$ ).

résolution V : les actions d'ordre I et IV ou II et III ont un risque d'alias (par exemple,  $P_1$  et  $P_2P_3P_4P_5$  ou  $P_1P_2$  et  $P_3P_4P_5$ ).

Les matrices d'Hadamard sont souvent utilisées avec une résolution III pour faire une analyse de sensibilité assez grossière et réduire le nombre de facteurs à traiter par la suite avec un plan de résolution supérieur.

#### b.6 Propriétés statistiques

*L'orthogonalité* est vérifiée quand la matrice variance-covariance  $(^T X.X)^{-1}$  est diagonale. Les matrices d'Hadamard, les plans complets à 2 niveaux et les plans fractionnaires à 2 niveaux respectent l'orthogonalité. Plus simplement : pour un niveau fixé de tout facteur du plan, chaque autre facteur apparaît un nombre égal de fois à chacun de ses niveaux possibles. On comprend ainsi intuitivement comment il est possible de définir l'effet d'un facteur dans une série d'expériences où tous les facteurs bougent en même temps : on "annule" l'effet des autres facteurs en choisissant les expériences dont on traite les résultats.

*L'isovariance* assure que la variance de la réponse (Y) est constante à distance fixe du centre du domaine, c'est le cas des matrices d'Hadamard, des plans complets à 2 niveaux et des plans fractionnaires à 2 niveaux [OZIL97].

*La précision uniforme* assure la même précision de Y estimée sur le domaine de validité du modèle.

Il existe enfin *les critères alphabétiques* (A, D, E, G, I...) portant à minimiser la variance des coefficients du modèle. Dans [CRARb01], [GIAN98] et [CRARa01], des MEMS sont optimisés à l'aide de plans I et D-optimaux.

#### b.7 Tables orthogonales de Taguchi

Il s'agit tout simplement de plans factoriels fractionnaires, respectant donc l'orthogonalité. L'intérêt de ces tables standard est leur facilité d'utilisation. Le Dr. Taguchi leur a associé différents outils permettant de choisir rapidement quel plan est le mieux adapté à un problème donné (*graphe des effets* permettant la prise en compte de la difficulté de modifier certains facteurs, *triangle des interactions* permettant de définir les alias) [PILL01]. Bien sur, en regardant de plus près parmi les nombreux plans existants, on pourrait peut être en trouver de mieux adaptés, mais les opérateurs du monde industriel où les concepteurs de MEMS ne possèdent a priori pas les connaissances statistiques pour choisir un plan de manière adéquate. Cela explique le succès des tables de Taguchi dans l'industrie.

Les tables orthogonales de Taguchi sont notées  $L_i(j^k)$  où  $i$  est le nombre d'expériences à réaliser,  $j$  le nombre de niveaux par facteur,  $k$  le nombre de facteurs. On peut remarquer que  $L_4(2^3)$ ,  $L_8(2^7)$ ,  $L_{12}(2^{11})$ ,  $L_{16}(2^{15})$  sont les matrices d'Hadamard  $H_4$ ,  $H_8$ ,  $H_{12}$  et  $H_{16}$ . En effet, les permutations de lignes (expériences) ou de colonnes (facteurs) ne changent pas les propriétés statistiques des plans. Dans ces cas, c'est toujours le modèle linéaire  $y(x) = b_0 + \sum_{i=1}^n a_i x_i$  qui est implicite, ces plans étant de résolution III, les actions du second ordre ou plus sont négligées.

### b.8 Analyse en Composantes Principales

Pour que le calcul des effets ne soit pas biaisé, il faut que les facteurs soient décorrélés. Or dans le cas des MEMS, certains paramètres de bruit (qui sont des "facteurs" pour les plans d'expériences) sont bien sûr corrélés, puisqu'il s'agit d'une technologie de type "intégrée". Par exemple, l'incertitude associée à la hauteur d'une poutre en polysilicium ( $X_1$ ) peut être liée à l'incertitude associée à l'épaisseur de la couche d'oxyde précédente ( $X_2$ ). Comme on peut le voir sur la figure [Figure 2-33], si la corrélation entre certains paramètres est forte, certaines combinaisons de valeurs n'ont pas de sens, puisqu'elles n'apparaîtront probablement jamais. Au lieu donc de considérer  $\{x_1, x_2\}$  nous allons effectuer un changement de base vers  $\{p_1, p_2\}$  qui sont des vecteurs orthonormés. Travailler sur ces vecteurs non corrélés ne biaisera pas les résultats en analyse de Monte Carlo ou dans les plans d'expériences. En réalisant ce changement de base, on obtient un nouvel espace de dimension identique.

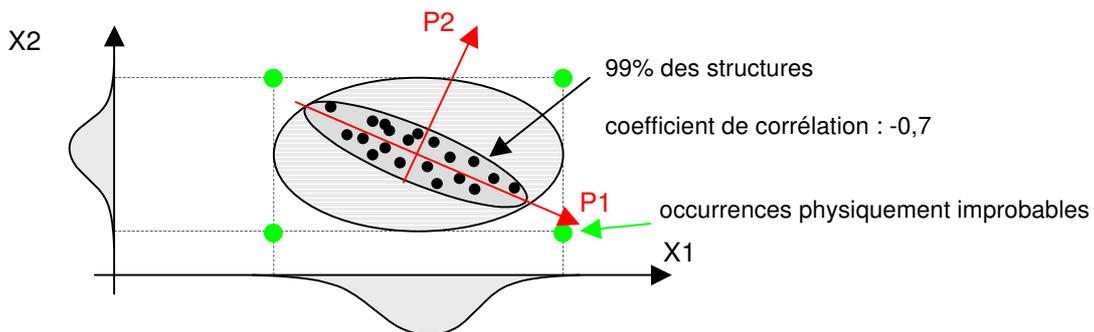


Figure 2-33 : Corrélation entre facteurs de bruit

Le changement de base effectué conserve la dimension du problème :

$$\begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_k \end{pmatrix} = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{k1} & \cdots & a_{kn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{pmatrix}.$$

Cette méthode s'appelle Analyse en Composantes Principales (ACP). Elle permet donc de s'assurer de la non corrélation de ces paramètres : il faut pour cela posséder les mesures réalisées en parallèle sur l'ensemble de ces paramètres. Utiliser les plans d'expériences sans ACP conduit à une estimation pessimiste du rendement et à une amélioration moins importante de ce dernier.

Souvent, l'ACP est suivie par une Analyse en Facteurs Communs (AFC) [CHEN93]. Cette méthode permet de mettre en évidence des facteurs sous-jacents influençant les variables originales. Son principal intérêt est la réduction de la dimension du nouvel espace ainsi obtenu. Les nouveaux facteurs sont définis par  $P_i = a_{i1}F_1 + a_{i2}F_2 + \dots + a_{ik}F_k + e_i$ , où  $P_i$  est une des variables originales,  $F_j$  est le  $j^{\text{ème}}$  facteur hypothétique, de valeur moyenne nulle et d'écart type unitaire. Les facteurs  $F$  sont décorrélés entre eux, et  $a_{ij}$  est appelé *facteur de charge*.  $e_i$  est l'erreur faite sur la  $i^{\text{ème}}$  variable, d'écart type  $\sigma_i$ . Un facteur  $F$  dont le facteur de charge serait faible (par comparaison aux autres) dans les expressions de chaque  $P_i$  pourrait être négligé sans grande influence sur la variance  $\sigma_i$  des  $P_i$ . On obtient ainsi une réduction du nombre de facteurs, tout en pouvant estimer la perte d'information sur la variance des facteurs initiaux.

#### b.9 Plans produits de Taguchi

Les plans produits sont obtenus en utilisant 2 tables orthogonales de Taguchi : l'une consacrée aux facteurs de dessin ( $x_i$ ,  $j$ , plan bruit) [Tableau 2-8]. On utilise des tables à 2 niveaux, car si nous ne sommes pas intéressés par les effets quadratiques mais par les interactions entre facteurs de bruit et de dessin (voir p. 88), c'est une configuration très pertinente : ces dernières sont implicitement prises en compte lors de l'utilisation de plans produits [PILL01].

| Exp. n°j | 1 | 2 | 3 | 4 |
|----------|---|---|---|---|
| 1        | - | - | + | + |
| 2        | - | + | - | + |
| 3        | - | + | + | - |

| Exp. n°i | x <sub>1</sub> | x <sub>2</sub> | x <sub>3</sub> |
|----------|----------------|----------------|----------------|
| 1        | -              | -              | -              |
| 2        | -              | +              | +              |
| 3        | +              | -              | +              |
| 4        | +              | +              | -              |

|                 |                 |                 |                 |
|-----------------|-----------------|-----------------|-----------------|
| Y <sub>11</sub> | Y <sub>12</sub> | Y <sub>13</sub> | Y <sub>14</sub> |
| Y <sub>21</sub> | Y <sub>22</sub> | Y <sub>23</sub> | Y <sub>24</sub> |
| Y <sub>31</sub> | Y <sub>32</sub> | Y <sub>33</sub> | Y <sub>34</sub> |
| Y <sub>41</sub> | Y <sub>42</sub> | Y <sub>43</sub> | Y <sub>44</sub> |

| Y              |                        |
|----------------|------------------------|
| Y <sub>1</sub> | $\bar{\sigma}_{1x_1-}$ |
| Y <sub>2</sub> | $\bar{\sigma}_{2x_1-}$ |
| Y <sub>3</sub> | $\bar{\sigma}_{3x_1+}$ |
| Y <sub>4</sub> | $\bar{\sigma}_{4x_1+}$ |

|                             |                       |
|-----------------------------|-----------------------|
| Y <sub>X<sub>1</sub>-</sub> | $\bar{\sigma}_{x_1-}$ |
| Y <sub>X<sub>1</sub>+</sub> | $\bar{\sigma}_{x_1+}$ |

**Tableau 2-8 : Plan produit de Taguchi**

On reconnaît dans ce tableau les 2 matrices des plans d'expériences. Chaque expérience du plan bruit sera répétée pour chaque expérience du plan principal. Nous aurons ainsi  $4 \cdot 4 = 16$  simulations à réaliser ( $Y_{ij}$ ). Comme on l'a vu pour le calcul des effets principaux (p. 71), on va s'intéresser aux résultats des simulations pour chaque niveau des facteurs  $x_i$ . Leur traitement sera différent selon que la performance est un scalaire (une fréquence de résonance par exemple) ou un vecteur (points d'une simulation AC par exemple). Dans le cas classique d'un scalaire, pour chaque expérience  $i$  du plan principal, on calcule la valeur moyenne  $Y_i$  de la performance ciblée :

$$Y_i = \frac{\sum_j Y_{ij}}{j_{\max}}, \text{ et sa variance : } \sigma_i^2 = \frac{\sum_j (Y_i - Y_{ij})^2}{j_{\max} - 1}. \text{ Pour connaître l'influence d'un facteur à un}$$

niveau donné, on moyennera les résultats de toutes les expériences où ce facteur est à ce

$$\text{niveau : } \bar{Y}_{x_i+} = \frac{\sum_i Y_{x_i+}}{i_{\max}/2} \text{ et } \sigma_{x_i+}^2 = \frac{\sum_i \sigma_{x_i+}^2}{i_{\max}/2}.$$

Dans le tableau, le cadre rouge foncé correspond aux simulations à considérer pour évaluer l'effet du facteur  $x_1$  au niveau haut (pointillés).

Dans le cas d'une performance cible qui serait un vecteur, l'approche est la même, mais les calculs légèrement différents, nous y reviendrons plus tard (p. 91).

Dans le monde de l'industrie, les plans produits sont lourds à réaliser, et ne le sont donc qu'une fois. Il faut ensuite choisir au vu des résultats quelle solution est la meilleure. Pour chaque combinaison de paramètres, on dispose de la valeur moyenne  $\mu$  et de la dispersion des performances  $\sigma$ . Il faut alors faire un choix, car il est peu probable que  $\mu$  soit optimale pour un  $\sigma$  minimum. Soit la meilleure solution est évidente, soit on peut s'aider des fonctions Pertes de Taguchi par exemple (p. 61) pour les comparer entre elles.

Les différents types de paramètres présents dans un problème d'optimisation statistique ont été définis. Nous avons exposé les principales méthodes d'estimation et d'optimisation du rendement et différentes fonctions visant à mesurer la qualité d'un système, ainsi que leurs avantages et inconvénients. Les méthodes sont nombreuses et peuvent être déclinées à l'envie. L'approche que nous avons choisie est donnée et justifiée dans le chapitre suivant.

## REFERENCES

- [AGIL-net] <http://eesof.tm.agilent.com/docs/adsdoc2002C/optstat/index.html>
- [ASPI-net] <http://www.mentor.com/ams/dfm.html#aspire>
- [BRAH84] Hachmi Braham, "Etudes de méthodologies en vue de l'optimisation statistique des performances des circuits", Th. : Elec. : Université Paul Sabatier, Toulouse : 1984 : 104p.
- [CHEN93] J. Chen, M.A. Styblinski, "A systematic approach of statistical modeling and its application to CMOS circuits", ISCAS 1993, Chicago, Volume 3, pp 1805-1808.
- [CRARa01] S.B. Crary et al., "Designing efficient computer experiments for metamodel generation", proceedings of Modeling and Simulation of Micosystems '01, Hilton Head Island, USA, mars 2001, pp. 132-135, disponible sur <http://www.cr.org>.
- [CRARb01] S.B. Crary, "Statistical design and analysis of computer experiments for the generation of parsimonious metamodels", DTIP 2001, proceedings of SPIE Vol. 4408, 2001, pp. 29-39.
- [DEAL97] Manuel de Almeida, "Etude et développement d'une méthode de simulation pour l'estimation des variations des performances des portes logiques CMOS à partir de l'analyse des fluctuations des procédés de fabrication", Th. : Elec. : Université Paris XI, Orsay : 1997 : 183 p.
- [DEBY98] Geert Debyser, Georges Gielen, "Efficient analog circuit synthesis with simultaneous yield an robustness optimizatoin", Proceedings of the 1998 IEEE/ACM international conference on Computer-aided design, 1998, San Jose, California-USA, 4 p.
- [DEVA94] Yann Deval, "Conception de circuits intégrés analogiques – Elaboration d'une méthode originale d'optimisation basée sur la techniques des plans d'expériences", Université Bordeaux I : 1994 : 177 p.
- [DIRE92] S.W. Director, P. Feldmann, K. Krishna, "Optimization of parametric yield: a tutorial", IEEE 1992 Custom Integrated Circuit Conference, pp 311-318.
- [DUBO87] Pierre Dubouix, "La conception optimale des circuits en présence de contraintes statistiques", Th. : Elec. : Université Paul Sabatier, Toulouse : 1987 : 175p.
- [GE99] P. Ge, N. Wang, "DOE study in building surrogate models for complex systems", Proceedings of DETC'00, ASME 2000, Baltimore, U.S.A.

- [GIAN98] Y.B. Gianchandani, S.B. Crary, "Parametric modeling of a microaccelerometer : comparing I and D-optimal design of experiments for finite-element analysis", IEEE Journal of Microelectromechanical Systems, Vol. 7, No. 2, juin 1998.
- [GIEL98] G. Gielen, G. Debyser, F. Leyn, W. Sansen, "Efficient yield estimation within automated analogue IC design", 1998 Workshop on Statistical Metrology Technical Papers, 0-7803-4338-7/98 IEEE, pp 118-121.
- [KERA97] Mansour Keramat, "Latin hypercube sampling Monte Carlo estimation of average quality index for integrated circuits", Analog Integrated Circuits and Signal Processing, 1997, Vol. 14, No. 1/2, pp. 131-142.
- [KERAA98] Mansour Keramat, "Analyse statistique et optimisation du rendement de fabrication des circuits électroniques", Th. : Elec. : Université de Paris Sud, Centre d'Orsay : 1998 : 124 p.
- [KERAb98] Mansour Keramat, "A study of stratified sampling in variance reduction techniques for parametric yield estimation", IEEE Trans. Circuits and Systems-II, Vol. 45 No. 5, mai 1998, 24 p.
- [LANS98] Lansberry, M.R., Jugulum, Rajesh, Ragsdell, K.M., "Some thoughts on the efficient calculation of the signal-to-noise ratio with application to the wheatstone bridge problem", American Supplier Institute, 1998, pp. 237-254.
- [LOW89] K.K. Low, S.W. Director, "An efficient methodology for building macromodels of IC fabrication processes", IEEE Transactions on CAD, Vol. 8 No. 12, December 1989, pp 1299-1313.
- [MA01] Lin Ma, E.K. Antonsson, "Robust mask-layout synthesis for MEMS", proceedings of Modeling and Simulation of Micosystems 2001, Hilton Head Island, U.S.A., pp. 128-131, disponible sur <http://www.cr.org>.
- [MILO90] L. Milor, A. Sangiovanni-Vincentelli, "Computing parametric yield accurately and efficiently", CH2924-9/90/0000/0116, 1990 IEEE, pp 116-119.
- [MOOR00] L.M. Moore, B.K. Ray, "Statistical methods for sensitivity and performance analysis in computer experiments", Proceedings of the 2000 Winter Simulation Conference, ed. J.A. Joines et al., pp. 486-491.
- [NAZZ00] D. Nazzal, M. Molaghasemi, L. Malone, "Evaluation of the effectiveness of group screening methods as compared to no group screening", Proceedings of the 2000 Winter Simulation Conference, ed. J.A. Joines et al., pp. 1499-1504.
- [OZIL97] Patrick Ozil, "Plan d'expériences (DOE), modélisation empirique, méthodologie de la surface de réponse, méthodologie Taguchi", ENS d'Electrochimie et d'Electrométallurgie, INPG, 1997, 161 p.
- [PAN93] Shao Wei Pan, Yu Hen Hu, "PYFS-A statistical Optimization method for integrated circuit yield enhancement", IEEE Transactions on CAD of IC and Systems, Vol 12. No. 12, February 1993, pp 296-309.

- [PETE01] D. Peters et al., "Enhanced optimization algorithms for the development of microsystems", DTIP 2001, proceedings of SPIE Vol. 4408, 2001, pp. 237-345.
- [PILL01] Maurice Pillet, "Les plans d'expériences par la méthode Taguchi", 3<sup>ème</sup> tirage 2001, Paris : Ed. d'Organisation, 330 p.
- [PLAC46] R.L. Plackett and J.P. Burman, "The design of optimum multifactorial experiments", 1946, Biometrika vol. 33, pp. 305-325.
- [POTT01] M.D. Pottenger, W.J. Kaiser, "Noise modeling methodology for full-system inertial microsensor codesign", proceedings of Modeling and Simulation of Microsystems 2001, Hilton Head Island, U.S.A., pp. 140-143, disponible sur <http://www.cr.org>.
- [SCHE01] F. Schenkel et al., "Mismatch analysis and direct yield optimization by spec-wise linearization and feasibility-guided search", DAC 2001, Las Vegas.
- [SCHW99] Robert Schwencker, "Automatic design centering of analog integrated circuit based on the Generalized Boundary Curve Multiple Robustness Objectives", 1999, rapport technique, Technische Universität München.
- [SMIT98] T.H. Smith et al., "Bias and variance in single and multiple response surface modeling", Third International Workshop on Statistical Metrology, IWSM 1998, Honolulu, 4 p.
- [STYB90] M.A. Styblinski, "Design for circuit quality: yield maximization, minimax, and Taguchi approach", CH2924-9/90/0000/0112, 1990 IEEE, pp 112-115.
- [STYB95] M.A. Styblinski, J.C. Zhang, "Yield and variability optimization of integrated circuits", Kluwer Academic Publishers, 1995, 234 p.
- [TROC00] Linda Trocine, L.C. Malone, "Finding important independent variables through screening designs: a comparison of methods", Proceedings of the 2000 Winter Simulation Conference, ed. J.A. Joines et al., pp. 749-754.
- [ZEBB96] Ahmed Zebbache, "Analyse et synthèse statistique des circuits électroniques : mise en œuvre du simulateur ouvert SPICE-PAC et de la méthode du recuit simulé", Th. : Elec. : Ecole Centrale Paris : 1996 : 105 p.



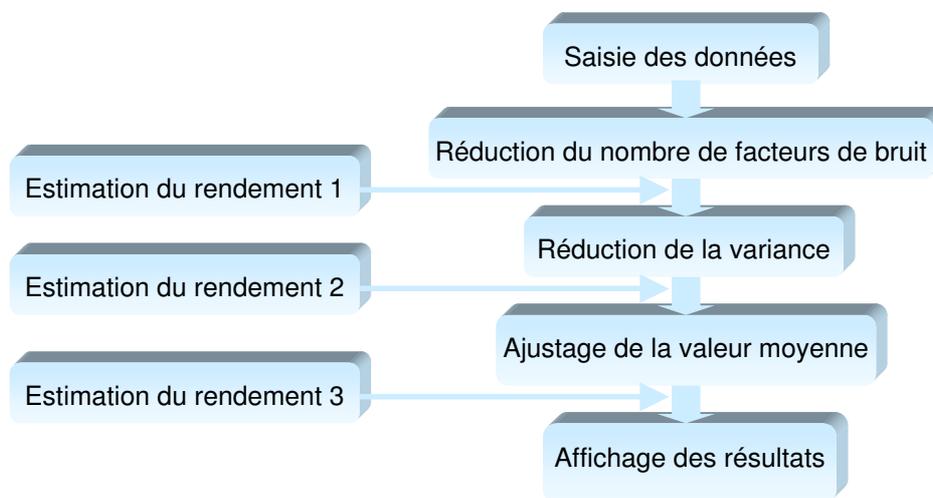
# Chapitre 3

## NOUVELLE STRATEGIE



### 3. NOUVELLE STRATEGIE

Comme nous l'avons vu dans le chapitre 2.4, les méthodes permettant d'améliorer le rendement sont nombreuses. Nous avons décidé de nous inspirer d'une méthode proposée dans [STYB95] (page 83), et qui utilise des plans d'expériences en deux temps : d'abord pour minimiser la variance de la performance ciblée, et dans un deuxième temps pour obtenir une réponse moyenne de bonne qualité (centrage du MEMS). Ces différents choix sont justifiés dans ce chapitre, et l'algorithme global donné par la Figure 3-1.



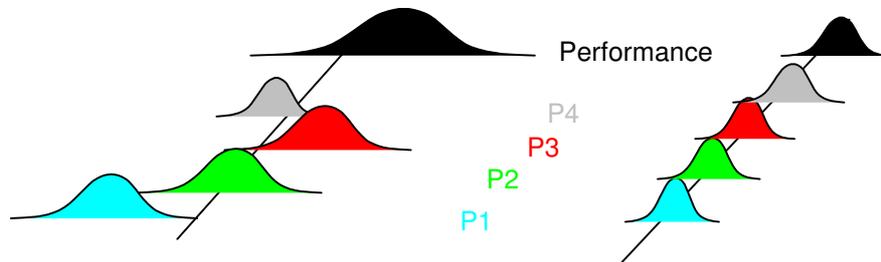
**Figure 3-1 : Organigramme de l'algorithme d'optimisation du rendement**

#### 3.1 JUSTIFICATION

##### 3.1.1 OPTIMISATION EN DEUX TEMPS

Il nous semble important de séparer les phases de réduction de la variance et de centrage du MEMS. On a vu que très souvent, la fonction optimisée dans le cadre de l'amélioration du rendement était soit la fonction Perte de Taguchi (EQL), soit plus souvent en microélectronique l'indice de capabilité (Cpk). Or, si ces fonctions prennent en compte à la fois la dispersion et la valeur moyenne de la réponse, elles ne le font pas de manière équitable. En effet, la valeur moyenne d'une réponse est beaucoup plus facile à modifier que sa variabilité, il n'est donc pas rare que les processus d'optimisation statistique du rendement paramétrique n'aient qu'un effet très faible sur la réduction de variance.

Un deuxième point est la jeunesse et la complexité des technologies MEMS qui contribuent à donner aux paramètres technologiques ( $p_i$ ), et donc aux performances, de fortes dispersions [Figure 3-2].



**Figure 3-2 : Une technologie jeune est pénalisante pour obtenir une faible dispersion des performances**

D'une manière générale, les tolérances rencontrées dans les circuits discrets sont plus faibles que pour les circuits intégrés, où les variations de la technologie ont plus d'impact. Comme elles sont mal contrôlées, on les compense souvent en imposant des symétries au niveau du circuit, comme pour réduire les problèmes dus aux variations de température, par exemple en utilisant des rapports de résistance ou de capacité. Le problème pour les MEMS est que la symétrie d'une structure n'est pas toujours envisageable. Encore une fois, il semble critique de s'intéresser de plus près à la réduction de variabilité.

Enfin, un dernier avantage de taille : un MEMS robuste (dessin centré, variabilité de la réponse réduite) minimise les risques d'avarie en cas de fonctionnement dans un milieu hostile (chocs, variations de température, d'humidité), et augmente logiquement son espérance de vie.

### 3.1.2 NATURE STATISTIQUE DU PROBLEME

Pour justifier l'utilisation des plans multiples, dont le but est bien sûr de réduire le nombre de simulations requises dans le processus d'optimisation, il faut regarder de plus près la nature statistique des paramètres.

#### a) Circuits discrets et circuit intégrés

Les algorithmes utilisés pour réduire la variabilité d'une performance dans le cadre de circuits discrets ne se sont pas avérés efficaces lorsqu'ils sont appliqués aux circuits intégrés. Les MEMS rencontreront le même problème, car l'origine en est la nature statistique des paramètres de bruit, différente selon que l'on a affaire à des assemblages discrets ou à des technologies de type "intégrées". On a vu (p. 70) que la performance ciblée d'un système peut être modélisée par un polynôme dont les termes sont les paramètres de dessin ( $x$ ) et de bruit ( $\theta$ ) du MEMS :

$$y(x,\theta) = b_0 + \sum_{i=1}^n a_i x_i + \sum_{j=1}^m d_j \theta_j + \sum_{i=1}^n \sum_{j=1}^m c_{ij} x_i \theta_j + \sum_{i=1}^n e_{ii} x_i^2 + \sum_{j=1}^m g_{jj} \theta_j^2 \quad (\text{éq. 3-1})$$

On ne va pas au-delà des termes quadratiques qui permettront de bien retranscrire une réponse éventuellement non linéaire. De même, on ne se préoccupe pas des interactions du troisième ordre ou plus car il est physiquement avéré qu'elles ne jouent presque jamais de rôle, même d'importance modérée. On s'intéresse à la réduction de la variabilité de la performance, on exprime donc sa variance :

$$\text{Var}(y) = \sigma_y^2 = E_\theta \left\{ (y - E(y))^2 \right\} = E_\theta (y^2) - \{E_\theta (y)\}^2 = \int_{R^m} [y(x,\theta) - \bar{y}(x)]^2 f_\theta(\theta) d\theta$$

avec  $\bar{y} = E_\theta \{y\} = \int_{R^m} y(x,\theta) f_\theta(\theta) d\theta$  (éq. 3-2), valeur moyenne de y et  $f_\theta(\theta)$  densité de probabilité de

On désire jouer sur cette variance en modifiant les paramètres de dessin ( $x_i$ ) :

$$\frac{\partial \text{Var}(y)}{\partial x_i} = \frac{\partial (E_\theta (y^2) - \{E_\theta (y)\}^2)}{\partial x_i} = 2 \int_{R^m} (y - \bar{y}) \frac{\partial y}{\partial x_i} f(\theta) d\theta$$

Or, comme  $\int_{R^m} (y - \bar{y}) f(\theta) d\theta = 0$  d'après (éq. 3-2), on en déduit que si  $\frac{\partial y}{\partial x_i}$  n'est pas une fonction de  $\theta$ , on ne pourra pas réduire la variance. Une condition nécessaire à la réduction

de variabilité est donc :  $\frac{\partial^2 y(x,\theta)}{\partial x_i \partial \theta_j} \neq 0$  (éq. 3-3)

### b) Cas des circuits discrets

Chaque paramètre de dessin ( $x_i$ ), par exemple une résistance, est associé à son propre paramètre de bruit ( $\theta_i$ ), qui traduit la tolérance sur cette résistance :

$$y(x,\theta) = y(x + \theta) \text{ et donc } \frac{\partial y}{\partial x_i} = \frac{\partial y}{\partial \theta_i} \quad (\text{éq. 3-4})$$

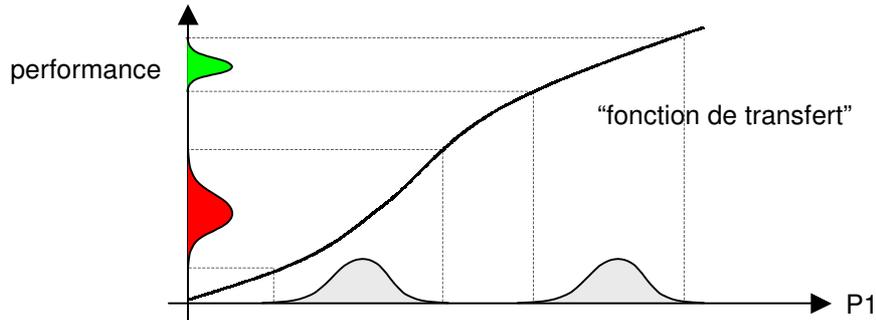
Des équations (éq. 3-3) et (éq. 3-4), on déduit que pour pouvoir minimiser la variance, il

faut vérifier :  $\frac{\partial^2 y(x + \theta)}{\partial x_i^2} \neq 0$  pour  $i=j$  et  $\frac{\partial^2 y(x,\theta)}{\partial x_i \partial x_j} \neq 0$  pour  $i \neq j$ .

On peut donc réduire la variance en jouant sur les paramètres possédant un coefficient important associé à leur terme quadratique [Figure 3-3], ainsi que sur les interactions. Mais les coefficients associés à ces dernières sont souvent beaucoup plus faibles que ceux des

termes quadratiques et les interactions sont aussi plus dures à contrôler : on ne travaillera donc que sur les termes quadratiques. La démarche sera donc généralement la suivante :

- ▶ La variance est réduite en optimisant les paramètres aux effets non linéaires sur la performance.
- ▶ La valeur moyenne est ajustée grâce aux paramètres aux effets linéaires importants et aux termes quadratiques faibles.



**Figure 3-3 : Effet des termes quadratiques sur la réduction de la variance**

### c) Cas des circuits intégrés

Chaque paramètre de dessin n'est plus associé à un paramètre de bruit particulier : l'espace vectoriel des paramètres de dessin n'est plus un sous-espace des paramètres de bruit. Un seul paramètre de bruit peut dégrader plusieurs paramètres de dessin. Par exemple, l'incertitude existant sur la largeur de grille en polysilicium d'une technologie donnée est, en première approximation, la même pour plusieurs transistors. On ne vérifie

donc plus l'équation (équ. 3-4)  $\frac{\partial y}{\partial x_i} = \frac{\partial y}{\partial \theta_i}$ , et l'équation (équ. 3-3)  $\frac{\partial^2 y(x, \theta)}{\partial x_i \partial \theta_j} \neq 0$  n'intervient

plus qu'à condition qu'il existe une interaction liant un paramètre de dessin et un paramètre de bruit, c'est à dire que l'influence d'un paramètre de bruit sur la performance ciblée du MEMS ne sera pas la même suivant la valeur du paramètre de dessin. Nous devons donc baser notre réduction de variance sur l'évaluation des interactions entre paramètres de dessin et de bruit. Le polynôme qui nous convient le mieux pour décrire la performance cible est donc linéaire et comporte les interactions adéquates  $(x_i \theta_j)$ . L'équation (équ. 3-1) débarrassée des termes quadratiques et considérée à proximité d'un point  $(y^0, x_i^0, \theta_j^0)$  s'écrira donc :

$$y(x, \theta) \approx y^0 + \sum_{i=1}^n a_i (x_i - x_i^0) + \sum_{j=1}^m d_j (\theta_j - \theta_j^0) + \sum_{i=1}^n \sum_{j=1}^m c_{ij} (x_i - x_i^0) (\theta_j - \theta_j^0)$$

En utilisant cette approximation, on peut démontrer [STYB95] (page 83) que le gradient de la variance et le coefficient de corrélation entre  $y$  et  $\theta_j$  s'approchent par :

$$\left. \frac{\partial \text{Var}(y)}{\partial x_i} \right|_{x=x_0} \approx 2 \sum_{j=1}^m c_{ij} d_j \sigma_j^2 \quad \text{et} \quad \left. \rho_{y,\theta_j} \right|_{x=x_0} = \frac{\text{CoVar}(y, \theta_j)}{\sigma_j \sigma_y} \approx d_j \frac{\sigma_j}{\sigma_y} \quad (\text{équ. 3-5})$$

La démarche à suivre dans le cas des circuits intégrés est donc la suivante :

- ▶ Réduire la variance en optimisant les paramètres  $x_i$  pour lesquels la somme  $\sum_{j=1}^m c_{ij} d_j \sigma_j^2$  est initialement importante.
- ▶ La valeur moyenne est ajustée grâce aux paramètres de dessin aux effets linéaires importants, et en dernier recours avec ceux qui étaient critiques lors de l'étape précédente.

### 3.1.3 UTILISATION DES PLANS MULTIPLES DE TAGUCHI

Puisque ce sont les interactions entre les facteurs de bruit et de dessin qui peuvent nous aider à réduire la variabilité de la réponse, nous nous intéressons aux plans produits de Taguchi qui semblent parfaitement adaptés.

Dans notre cas, ce n'est pas une performance qui s'exprime sous forme de scalaire que l'on aimerait pouvoir optimiser pour nos MEMS dans un premier temps, mais une performance sous forme de vecteur. Il faut donc adapter le calcul de la dispersion et de la valeur moyenne déductibles des plans produits.

Prenons l'exemple de  $x_1$ , au niveau haut pour la troisième expérience. Que pouvons-nous considérer comme valeur moyenne et comme dispersion dans le cas où l'on dispose d'un vecteur de dimension  $n$  ? Il va falloir les redéfinir, car

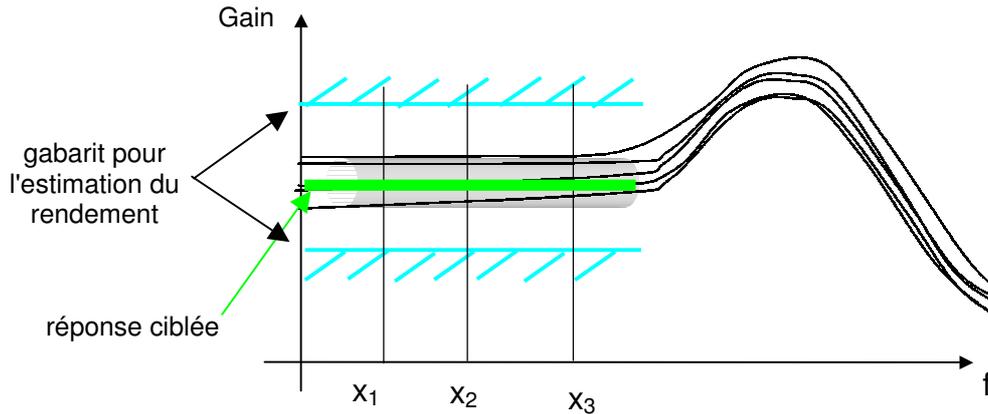
correspond maintenant à 

|          |          |          |          |
|----------|----------|----------|----------|
| $Y_{31}$ | $Y_{32}$ | $Y_{33}$ | $Y_{34}$ |
|----------|----------|----------|----------|

 .

|            |            |            |            |
|------------|------------|------------|------------|
| $Y_{31_1}$ | $Y_{32_1}$ | $Y_{33_1}$ | $Y_{34_1}$ |
| $Y_{31_2}$ | $Y_{32_2}$ | $Y_{33_2}$ | $Y_{34_2}$ |
| ...        | ...        | ...        | ...        |
| $Y_{31_n}$ | $Y_{32_n}$ | $Y_{33_n}$ | $Y_{34_n}$ |

Les calculs ne peuvent donc plus s'effectuer de la même manière. Nous sommes particulièrement intéressés par les réponses en analyse AC des MEMS. Le problème est représenté sur la [Figure 3-4].

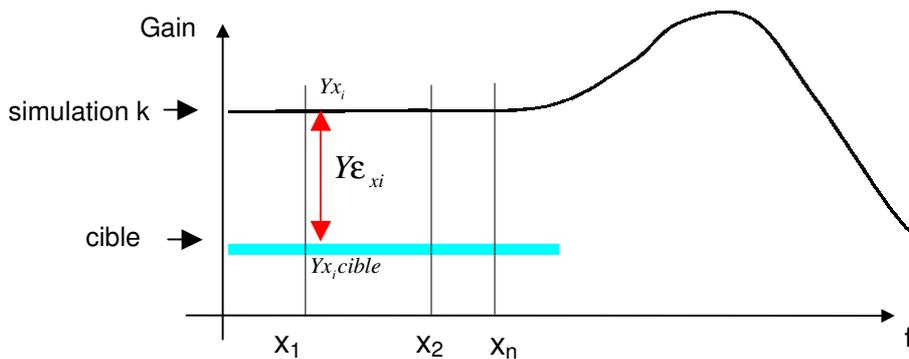


**Figure 3-4 : Simulations AC d'un passe-bas en présence de bruit**

Les lignes hachées représentent le gabarit que l'on utilisera pour estimer le rendement: une simulation sera plus tard jugée hors spécifications si l'un de ses points n'est pas compris dans l'intervalle défini par le gabarit. La ligne épaisse au centre représente la valeur vers laquelle on veut tendre lors de l'optimisation. Le tube donne une idée de la dispersion que l'on voit apparaître si l'on réalise de multiples analyses dans des conditions de bruit différentes.

**a) Calcul de la "moyenne"**

Il s'agit en réalité maintenant d'une fonction erreur que nous évaluons en utilisant la méthode des moindres carrés. Optimiser la performance ciblée reviendra par la suite à minimiser cette erreur, représentée sur la [Figure 3-5].



**Figure 3-5 : Calcul de l'erreur sur une simulation**

On peut calculer l'erreur de cette simulation vis-à-vis de la valeur ciblée avec une

fonction erreur, par exemple les moindres carrés : 
$$\epsilon_{simu} = \frac{1}{n} \sum_{x_i} \epsilon_{x_i} = \frac{1}{n} \sum_{x_i} (Yx_i - Yx_i,cible)^2$$

**b) Calcul de la dispersion**

Pour la calculer, on considère toutes les simulations réalisées pour une combinaison i (ici i=3) du plan principal :

|                  |                  |                  |                  |  |
|------------------|------------------|------------------|------------------|--|
| Y31 <sub>1</sub> | Y32 <sub>1</sub> | Y33 <sub>1</sub> | Y34 <sub>1</sub> | $\sigma_{3_1} = \frac{1}{j_{\max} - 1} \sum \left( Y_{3j_1} - \bar{Y}_{3_1} \right)^2$ |
| Y31 <sub>2</sub> | Y32 <sub>2</sub> | Y33 <sub>2</sub> | Y34 <sub>2</sub> | $\sigma_{3_2}$   |
| ...              | ...              | ...              | ...              | ...  |
| Y31 <sub>n</sub> | Y32 <sub>n</sub> | Y33 <sub>n</sub> | Y34 <sub>n</sub> | $\sigma_{3_n}$   |
|                  |                  |                  |                  | $\bar{\sigma}_3 = \frac{1}{n} \sum_{i=1}^n \sigma_{3_i}$                               |

On remarque que le calcul de la dispersion pour un point x<sub>i</sub> nécessite auparavant de calculer la réponse moyenne en ce point de toutes les simulations de bruit effectuées pour une combinaison du plan principal. On peut réécrire le plan produit donné lorsque la cible est un scalaire [Tableau 3-1], mais ce n'est plus la réponse moyenne que l'on essaie d'optimiser, mais une fonction erreur que l'on essaie de réduire.

|  |  |  |  |          |   |   |   |   |
|--|--|--|--|----------|---|---|---|---|
|  |  |  |  | Exp. n°j | 1 | 2 | 3 | 4 |
|  |  |  |  | 1        | - | - | + | + |
|  |  |  |  | 2        | - | + | - | + |
|  |  |  |  | 3        | - | + | + | - |

|          |                |                |                |
|----------|----------------|----------------|----------------|
| Exp. n°i | x <sub>1</sub> | x <sub>2</sub> | x <sub>3</sub> |
| 1        | -              | -              | -              |
| 2        | -              | +              | +              |
| 3        | +              | -              | +              |
| 4        | +              | +              | -              |

|                 |                 |                 |                 |
|-----------------|-----------------|-----------------|-----------------|
| Y <sub>11</sub> | Y <sub>12</sub> | Y <sub>13</sub> | Y <sub>14</sub> |
| Y <sub>21</sub> | Y <sub>22</sub> | Y <sub>23</sub> | Y <sub>24</sub> |
| Y <sub>31</sub> | Y <sub>32</sub> | Y <sub>33</sub> | Y <sub>34</sub> |
| Y <sub>41</sub> | Y <sub>42</sub> | Y <sub>43</sub> | Y <sub>44</sub> |

|                    |                  |
|--------------------|------------------|
| $\bar{\epsilon}_1$ | $\bar{\sigma}_1$ |
| $\bar{\epsilon}_2$ | $\bar{\sigma}_2$ |
| $\bar{\epsilon}_3$ | $\bar{\sigma}_3$ |
| $\bar{\epsilon}_4$ | $\bar{\sigma}_4$ |

|                         |                       |
|-------------------------|-----------------------|
| $\bar{\epsilon}_{x_1-}$ | $\bar{\sigma}_{x_1-}$ |
| $\bar{\epsilon}_{x_1+}$ | $\bar{\sigma}_{x_1+}$ |

**Tableau 3-1 : Plan produit : cas où la cible est un vecteur**

Si l'on s'intéresse à l'effet du facteur x<sub>1</sub> au niveau haut (pointillé), on calcule comme nous venons de le voir  $\bar{\sigma}_3$  et  $\bar{\sigma}_4$ , puis leur moyenne  $\bar{\sigma}_{x_1+}$ .

De même, on calcule notre fonction erreur pour chaque simulation, puis on la moyenne sur chaque combinaison du plan principal :  $\bar{\varepsilon}_1$ ,  $\bar{\varepsilon}_2$ ,  $\bar{\varepsilon}_3$ ,  $\bar{\varepsilon}_4$ , avant de la moyenner à nouveau pour le résultat final, en ne considérant que les expériences adéquates du plan principal.

### **3.2 SPECIFICATIONS ET PERFORMANCES**

Les spécifications seront données sous forme de limites et de gabarit (Figure 2-1) dans un premier temps. Une fois l'algorithme global pleinement validé, il sera très aisé de proposer une alternative telle que de fournir une fonction pour décrire une limite ou un courbe cible sur un intervalle donné. Cela ne fera que faciliter la saisie du problème, mais ne changera donc en rien sa méthode de résolution.

Concernant l'évaluation des performances, nous considérerons de même le cas le plus simple : une courbe de référence pour des analyses transitoires et surtout AC.

### **3.3 ESTIMATION DU RENDEMENT**

Nous avons vu (§ 2.3) qu'il y avait différentes façons d'évaluer le rendement par des méthodes de type Monte-Carlo. Y consacrer du temps n'est pas superflu puisque l'on peut réduire la variance sur l'estimateur de Monte-Carlo tout en réduisant le nombre de simulations requises. Toutefois, notre but n'est pas de travailler sur ce problème, mais de nous consacrer seulement à l'optimisation du rendement. Nous utiliserons donc une analyse de Monte-Carlo classique pour réaliser des tirages aléatoires, la plus recommandée lorsque l'on ne dispose pas d'une fonction de performance. Ce critère n'entrera donc pas en considération lors de l'évaluation globale de l'algorithme.

### **3.4 REDUCTION DE LA VARIABILITE D'UNE PERFORMANCE**

Nous avons vu que pour réduire la variabilité, il fallait s'intéresser aux interactions entre les facteurs de dessin et les facteurs de bruit. On désire utiliser un algorithme de forme itérative : on ne cherche pas quelle est la solution optimale, c'est à dire la combinaison de valeurs des facteurs qui minimiserait la dispersion de la performance, mais sa direction. A chaque nouvelle itération, chaque facteur évoluera donc suivant cette direction. Pour la connaître, nous utilisons les plans produits qui peuvent correspondre implicitement à un modèle factoriel (équ. 2-3) qui comporte ces interactions. Nous pouvons choisir une bonne résolution lorsque l'on dispose de peu de facteurs de dessins (V ou plan complet) mais dans le cas contraire, nous choisirons des plans de résolution III. Dans le cas où il existerait une forte interaction entre deux paramètres de dessin, on risque d'attribuer par la suite à un

facteur de dessin (qui lui est aliasé) une faculté à minimiser la variance de la réponse qu'il ne possède peut être pas. Le facteur sera donc conservé lors de l'itération suivante, alors que son éventuelle élimination pourrait nous faire l'économie de simulations, même si elles ne sont jamais vraiment perdues. Pour limiter cet inconvénient, nous n'utiliserons des plans de si faible résolution que pour un nombre de facteurs de dessin supérieur à 16.

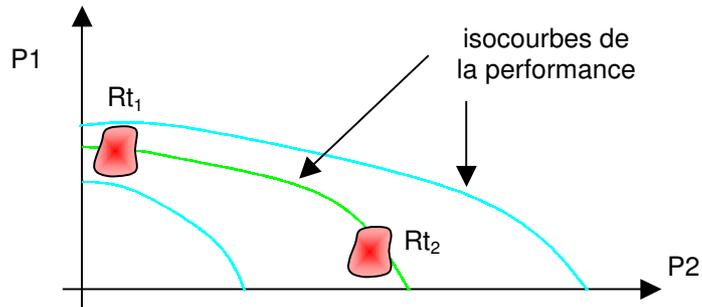
### 3.4.1 REDUCTION PREALABLE DU NOMBRE DE FACTEURS

Puisque nous utilisons les plans produits de Taguchi, le nombre d'expériences à réaliser à chaque itération est le produit du nombre d'expériences du plan principal et du plan bruit (i,j). Déterminer les interactions pour chaque couple  $(x_i, \theta_j)$  peut être très onéreux en nombre de simulations. Il est donc très rentable de réaliser un premier plan ne comportant que des facteurs de bruit afin de déterminer leur importance pour la réduction de variabilité du MEMS. On évalue ainsi leur coefficient de corrélation (équ. 3-5) et on élimine les facteurs de bruit dont la variabilité n'influe pas sur celle de la performance. Ces facteurs seront réintégrés par la suite pour les phases d'estimation du rendement et d'ajustage de la valeur moyenne de la performance, où ils ont un autre rôle à jouer.

Concernant les facteurs de dessin, cela est plus délicat : en prenant un plan unique sans considérer les facteurs de bruit, on ne peut juger à leur effet principal s'ils ont un rôle à jouer dans la réduction de variabilité (équ. 3-3). Il peut donc être dommageable pour cette phase de supprimer d'emblée du processus d'optimisation certains facteurs de dessin qui sembleraient peu important.

### 3.4.2 ALGORITHME DE REDUCTION DE LA VARIABILITE

Notre choix a donc porté sur l'utilisation des plans produits de manière itérative. A l'issue de la réalisation des expériences, en l'occurrence des simulations, nous effectuons les calculs (p. 93) qui nous permettent d'évaluer l'impact du choix des niveaux des facteurs de dessin sur la dispersion de la performance. Nous déterminons ainsi une direction dans laquelle il semble préférable de faire évoluer chaque facteur de dessin afin de réduire la variance de la réponse [Figure 3-6].

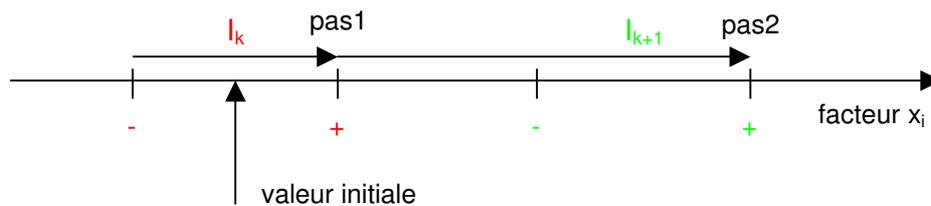


**Figure 3-6 : Déplacer la région de tolérance pour minimiser la dispersion**

Lors de l'itération suivante, les facteurs se voient attribuer deux nouveaux niveaux qui devraient à la fois fournir une variance moindre qu'à l'itération précédente et donner une indication sur les valeurs des niveaux à choisir pour l'itération suivante.

**a) Pas d'incrémentation**

Il est critique dans la plupart des algorithmes. En effet, trop petit, il y a un risque de ne pas voir d'amélioration notable d'un niveau à l'autre, ou de nous faire tomber dans un minimum local. Trop grand, nous prenons le risque de ne pas voir des minimums qui seraient très satisfaisants. Souvent, quand le pas est imposé dès le départ de manière fixe, on le prend comme un pourcentage de la valeur nominale du facteur auquel il s'applique [Figure 3-7]. C'est la première approche que nous avons choisie pour nous concentrer sur d'autres aspects du problème. Nous pouvons distinguer deux pas : *pas1* qui sépare les niveaux bas et haut d'un facteur d'un plan d'expériences, et *pas2* qui sépare les deux mêmes niveaux d'un facteur lorsque l'on considère deux itérations ( $I_k$ ) successives.



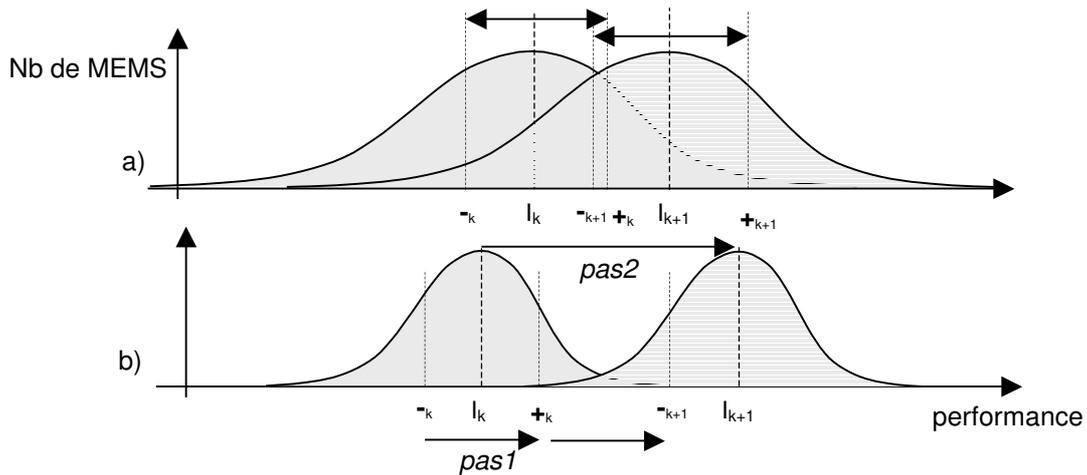
**Figure 3-7 : Pas d'incrémentation**

On peut très bien envisager la réalisation d'un plan produit initial avec un pas pondéré par la connaissance que le concepteur a du MEMS, puis le reconsidérer automatiquement à chaque itération en fonction des effets principaux. Pour un effet principal important, il semble censé de diminuer le pas et à l'inverse, d'augmenter l'écart entre les niveaux des facteurs qui semblent de moindre importance.

Le pas d'incrémentation doit aussi tenir compte, dans le cas des paramètres mixtes, de la dispersion propre au paramètre, pour ne pas confondre l'effet réel du à l'augmentation de sa valeur avec du bruit, comme sur la Figure 3-8 a. Cela nous amène à définir une relation

entre *pas1* et *pas2*. Dans l'algorithme, *pas1* est appelé *pas*, et nous considérerons tout d'abord que

*pas2=2*. [Figure 3-8 b]. De la sorte, le risque de confondre bruit et effet du facteur est minimisé et l'on conserve le même écart entre deux niveaux d'une même itération et les niveaux adjacents de deux itérations successives. Lors de la mise au point de l'algorithme, cela simplifie la compréhension des résultats.



**Figure 3-8 : Détermination d'un pas minimum**

Il s'est avéré par la suite que pour une dizaine de paramètres mixtes, un pas initial de être choisi sans que cela entraîne obligatoirement une confusion notoire quant à la direction à suivre pour l'itération suivante. Le pas minimum dépend beaucoup du nombre de paramètres de bruit et de leur influence respective sur la variabilité globale de la réponse. Une analyse de sensibilité réalisée avec les premières itérations permettrait de mieux choisir le pas.

### **b) Diminuer le nombre de facteurs**

Le but étant évidemment de réduire le nombre de simulations à réaliser. Chaque fois qu'un facteur est fixé, il disparaît du plan principal et prend la valeur du niveau qui minimise au mieux la variabilité. Nous avons identifié cinq points critiques pour diminuer petit à petit la dimension du problème :

#### **Trop faible amélioration due à un facteur :**

Si les deux niveaux d'un facteur ne se distinguent pas par une variation notable de la dispersion de la réponse, c'est qu'il n'est guère intéressant de le conserver dans le processus d'optimisation, nous le fixons donc au niveau le plus avantageux, et nous le retirons du plan principal. Soient  $\sigma_1$  et  $\sigma_2$  les valeurs de la dispersion associées aux niveaux

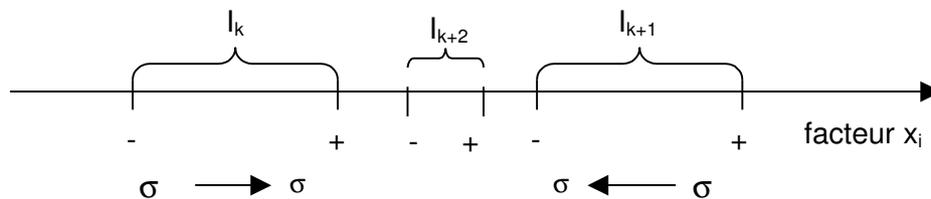
bas et haut d'un facteur de dessin.  $Ps_1$  est un pourcentage fixé par l'utilisateur tel que si  $|\sigma_2 - \sigma_1| < Ps_1 \max[\sigma_2, \sigma_1]$ , alors le facteur considéré est fixé.

**Surnombre des facteurs de dessin :**

Pour ramener rapidement un problème de forte dimension à un nombre de facteurs plus raisonnable, nous pouvons choisir quel sera le pourcentage d'entre eux automatiquement fixé à chaque nouvelle itération. Les facteurs sont classés suivant l'ordre de grandeur de la différence de la dispersion entre leur niveau haut et bas ( $|\sigma_2 - \sigma_1|$ ).  $Ps_2$  est le pourcentage de facteur à fixer, il y en a donc  $Ps_2 \cdot \text{nombre de facteurs}$ . Les facteurs sont rangés par ordre décroissant suivant  $|\sigma_2 - \sigma_1|$ , ce sont donc les derniers de la liste que l'on fixe en premier.

**Valeur optimale cernée :**

Il y a un travail important à effectuer sur le pas d'incrémentation, mais pour une première approche à pas fixe, on peut faire une simple amélioration lorsque la direction optimale préconisée est modifiée d'une itération  $I_k$  à la suivante,  $I_{k+1}$ . Sur la Figure 3-9, on peut voir que les prochains niveaux de la dernière itération pour ce facteur avant qu'il ne soit fixé sont choisis entre les niveaux les plus prometteurs des deux dernières itérations. Le facteur sera ensuite fixé au meilleur des quatre niveaux.



**Figure 3-9 : Oscillations autour d'un minimum**

Il y a donc des risques de tomber dans des minimum locaux si la variabilité n'est pas monotone dans l'intervalle.

**Dépassement des valeurs limites :**

On est aussi amené à fixer un facteur quand ce dernier a atteint les limites de son intervalle de validité. Comme dans le cas précédent, le pas est alors légèrement modifié pour être plus intéressant [Figure 3-10], et le facteur est fixé au meilleur des quatre derniers niveaux.

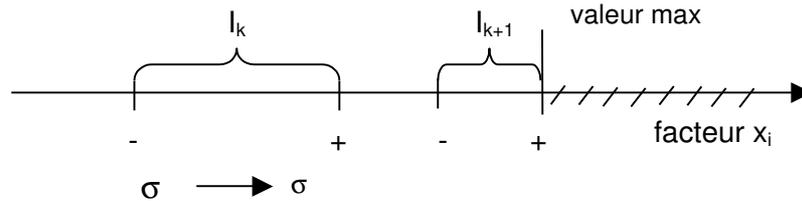


Figure 3-10 : Valeur limite atteinte pour un facteur de dessin

### Dégradation de la valeur moyenne de la réponse :

Lorsque l'on fait varier la valeur des paramètres d'un MEMS pour réduire la variance d'une performance, il se peut que l'on dégrade fortement la valeur moyenne de cette performance. Pour limiter cet effet négatif, on introduit un troisième paramètre,  $Ps_3$ . On fixera ainsi les facteurs qui ne réduisent la variance qu'en dégradant fortement la valeur moyenne de la performance et qui vérifient  $\Delta\varepsilon \cdot \Delta\sigma < 0$  et  $\Delta\varepsilon > Ps_3 \max[\varepsilon_2, \varepsilon_1]$ .

#### c) Facteurs non fixés

Les facteurs qui ne sont pas fixés ont a priori un effet important. Si nous réalisons de vraies expériences et non des simulations, il faudrait s'en assurer en faisant une *analyse de la variance*. En effet, cela ne nécessite pas de simulation supplémentaire, mais cela permet de s'assurer que l'amélioration que nous estimons significative n'est pas due à un simple effet de la variabilité des performances due aux facteurs de bruit non contrôlés, c'est à dire non présents dans le plan. Dans le cas de nos simulations, les éventuels facteurs de bruit non contrôlés sont fixés, l'analyse de variance n'est donc pas nécessaire.

#### d) Critères d'arrêt

Le nombre d'itérations maximum peut être fixé (boucleAmax), mais la phase de réduction de la variance se finit aussi dans le cas où tous les facteurs se retrouvent fixés avant d'atteindre cette limite.

#### e) Organigramme

Nous pouvons regrouper tous les cas et critères énoncés dans un organigramme [Figure 3-11]. Les niveaux sont notés N, 1 pour bas, 2 pour haut, min ou max pour leurs valeurs limites. Si l'on se réfère aux valeurs de l'itération précédente, on ajoute ' après les variables. Les cases carrées sont des tests. FinBoucle est un booléen indiquant s'il faut sortir de la boucle ou fixer le facteur après la prochaine itération. FixFact est un booléen stipulant que le facteur concerné doit être fixé, après la prochaine itération, au meilleur niveau parmi  $N_1, N_2, N_1'$  et  $N_2'$ .

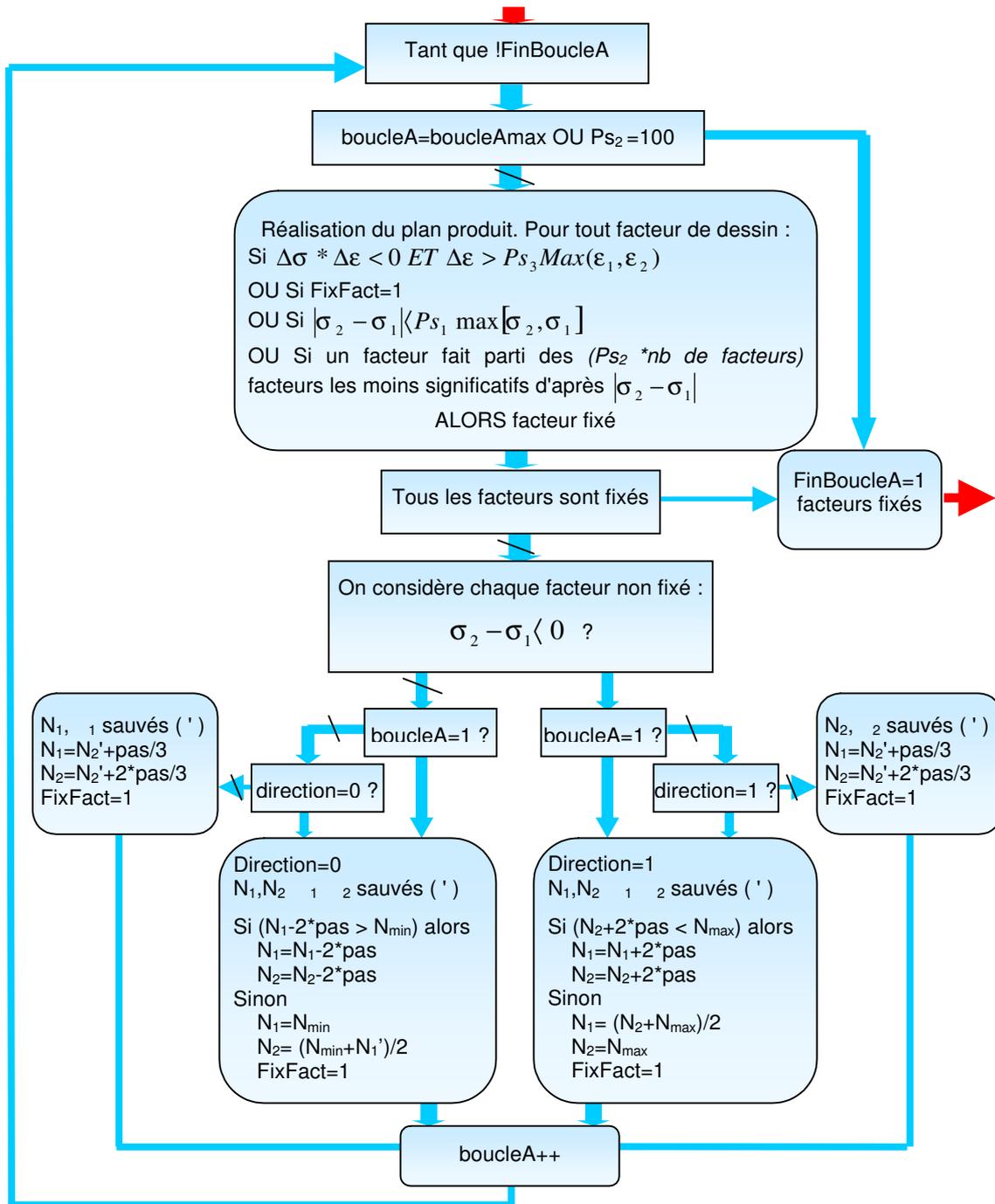


Figure 3-11 : Organigramme de la réduction de la variabilité

### 3.5 AJUSTEMENT DE LA REPONSE MOYENNE DU SYSTEME

Il se peut que l'étape de réduction de variabilité nous ait rapproché de la valeur cible pour la valeur moyenne de la performance du MEMS. Toutefois, ce cas reste marginal et généralement, la valeur moyenne de la mesure de la performance est fortement dégradée. Nous procédons donc à la deuxième phase de notre processus d'optimisation : l'ajustement de la réponse moyenne du MEMS.

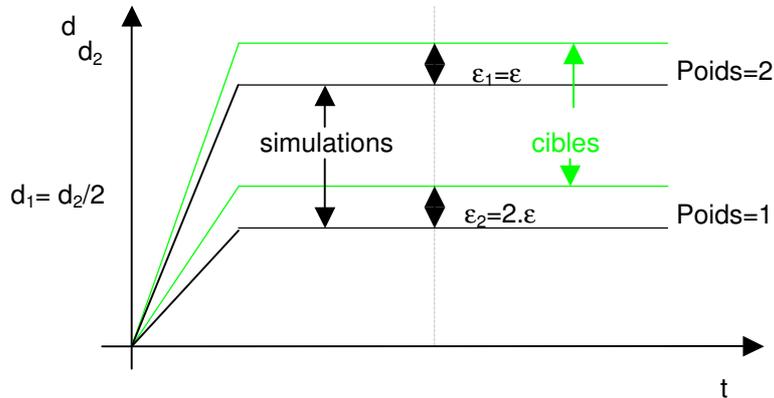
Nous reprenons à nouveau l'ensemble des facteurs de bruit en compte. En effet, ceux qui avaient été écartés du processus d'optimisation car ils ne pouvaient pas contribuer à l'étape de réduction de la variance doivent par contre être considérés pour la réduction de l'erreur sur la valeur moyenne de la performance.

### 3.5.1 OPTIMISATION MULTI-OBJECTIFS : ANTICIPER LES PROBLEMES A VENIR

Avant de définir la fonction coût retenue, il est utile de se pencher sur la prochaine étape logique pour laquelle elle sera cruciale : l'optimisation multi-objectifs. Il s'agira non plus d'optimiser une performance du MEMS, mais un ensemble de performances de natures éventuellement très différentes : déplacement maximum, variation de capacité, déformation, fréquence de résonance... Pour pouvoir comparer les erreurs associées à ces performances, la première approche est de s'affranchir de l'unité et de les ramener à un ordre de grandeur comparable. En tout point, cette opération est réalisée simplement par une division de l'erreur absolue (différence en un point entre la simulation et la cible) par  $Y_{ref}$ , obtenant ainsi une erreur relative.  $Y_{ref}$  doit être identique d'une simulation à l'autre pour juger de l'amélioration apportée par telle ou telle combinaison de paramètre.  $Y_{ref}$  doit donc être fixé non pas en fonction de la simulation mais des contraintes d'optimisation sur la performance traitée. Classiquement,  $Y_{ref} = \frac{1}{n} \sum_{i=1}^n |Y_{x_i, cible}|$ . Cette solution n'est pas idéale, notamment lorsque  $Y_{ref}$  est proche de 0 ou que l'on dispose de peu de points définissant la courbe ciblée, mais c'est une première approche qui donne de bons résultats dans la plupart des cas. Par ailleurs, aucune solution globale ne semble être proposée dans la littérature pour aborder au mieux ce problème. Dans le cas où une seule performance intervient, nous serions tentés de fixer  $Y_{ref}$  à 1 mais l'erreur en un point dans les spécifications risquerait d'être plus grande que lorsque l'on est tout juste hors spécifications en ce point, suivant les fonctions coût considérées. On rencontre aussi un problème si l'on a affaire à une courbe cible dont le minimum et le maximum sont très éloignés et où seuls un petit nombre de points sont à ces extrémités et n'influencent pas  $Y_{ref}$ .

Le problème de considérer une erreur relative dans le cadre d'une optimisation multi-objectifs est que l'on attribuera une importance différente à l'erreur en un point suivant l'amplitude de la réponse en ce point. Cela peut être un effet non désirable, comme on le voit sur la Figure 3-12 où l'on souhaite des déplacements  $d_1$  et  $d_2$  différents pour 2 poutres, mais sur lesquels on admet les mêmes incertitudes ou tolérances. Hors, de par la formulation de l'erreur, l'algorithme privilégiera la réduction de l'erreur du déplacement le moins important.

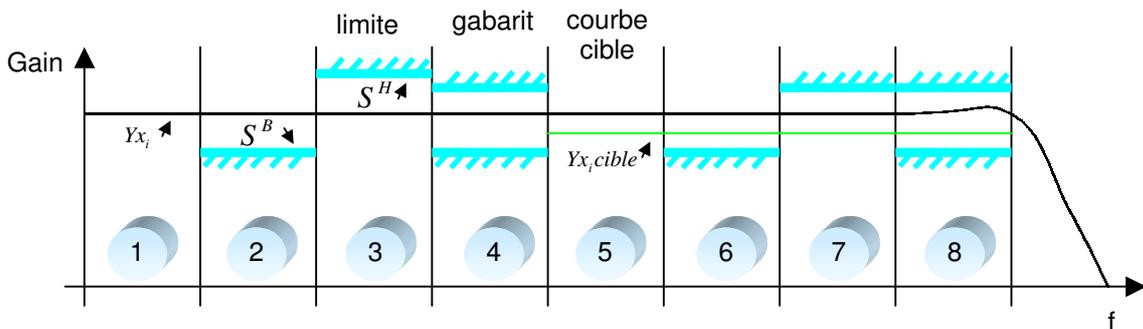
Pour contrebalancer cet effet, on peut associer un poids plus important à la performance "d1", mais cela reste un problème de fond.



**Figure 3-12 : L'erreur est proportionnelle à l'amplitude de la réponse**

### 3.5.2 CHOIX DE LA FONCTION ERREUR

La fonction erreur choisie doit pouvoir tenir compte de différents cas. Nous prenons un exemple pour une analyse AC d'un MEMS dont le comportement en fréquence est analogue à celui d'un filtre passe-bas [Figure 3-13]. La fonction erreur doit retranscrire tous les cas qui peuvent se présenter en un point suivant la contrainte qui y est associée : un gabarit, une limite supérieure ou inférieure, une courbe cible, une combinaison de ces possibilités ou rien. La figure suivante reprend tous les cas envisagés.



**Figure 3-13 : Combinaison de contraintes pour une optimisation des performances**

Dans chacun de ces cas, la fonction erreur est fixée. Ce choix initial pourra encore être affiné avec l'expérience et la confrontation avec des cas particuliers. Pour un point (ici une fréquence) où les limites sont respectées, l'erreur associée est la suivante :

1.  $\varepsilon_{x_i} = 0$ , puisque aucune contrainte n'est fixée, aucune erreur n'est comptabilisée pour ce point.

2.  $\varepsilon_{x_i} = e^{-\left(\frac{Y_{x_i} - S^B}{Y_{ref}}\right)}$ . L'exponentielle négative permet une erreur inférieure à 1 et rapidement décroissante lorsque l'on s'éloigne de la limite. Pour le calcul de  $Y_{ref}$ , on prendra  $Y_{x_i,cible} = S^B$  en ce point.

3.  $\varepsilon_{x_i} = e^{-\left(\frac{S^H - Y_{x_i}}{Y_{ref}}\right)}$ . On prendra ici  $Y_{x_i,cible} = S^H$ .

4.  $\varepsilon_{x_i} = 0,25 \cdot \left(\frac{Y_{x_i,cible} - Y_{x_i}}{Y_{ref}}\right)^2$ . On prendra ici  $Y_{x_i,cible} = \left(\frac{S^H - S^B}{2}\right)$ .

Pour les cas 5 à 8, nous disposons d'une courbe cible : nous utiliserons donc les moindres carrés et  $Y_{x_i,cible}$ , qui est fourni, pour le calcul de  $Y_{ref}$ .

5.  $\varepsilon_{x_i} = 0,5 \cdot \left(\frac{Y_{x_i} - Y_{x_i,cible}}{Y_{ref}}\right)^2$ .

6.  $\varepsilon_{x_i} = 0,75 \cdot \left(\frac{Y_{x_i} - Y_{x_i,cible}}{Y_{ref}}\right)^2$  pour tenir compte du danger supplémentaire du à  $S^B$ .

7.  $\varepsilon_{x_i} = 0,75 \cdot \left(\frac{Y_{x_i} - Y_{x_i,cible}}{Y_{ref}}\right)^2$  pour tenir compte du danger supplémentaire du à  $S^H$ .

8.  $\varepsilon_{x_i} = 1 \cdot \left(\frac{Y_{x_i} - Y_{x_i,cible}}{Y_{ref}}\right)^2$  pour tenir compte de la présence d'un gabarit.

Dans la pratique, il vaut mieux éviter de mélanger les types de fonctions d'erreurs (moindres carrés et exponentielles). C'est pourquoi lorsqu'il sera nécessaire de définir une courbe cible, il est préférable de le faire en tous points où il existe des contraintes si cela est possible, pour ne pas avoir recours aux exponentielles. A contrario, si seules les limites suffisent au problème d'optimisation et qu'il n'est pas absolument nécessaire d'utiliser une courbe cible, nous en ferons l'économie. Les coefficients présents devant les expressions peuvent être ajustés en présence de cas particuliers, le problème étant d'en prévoir le plus grand nombre avant d'y être confronté pratiquement.

Ces fonctions d'erreurs sont considérées dans le cas où les spécifications sont respectées. Mais lorsqu'un point s'avère hors spécifications, il faut donner plus de poids à l'erreur qui lui est associée. La plupart des erreurs définies précédemment sont en pratique

généralement inférieures à 1. La fonction erreur pour un point hors spécifications doit donc être rapidement supérieur à 1. Nous prendrons donc l'exponentielle positive :

- si  $Y_{x_i} > S^H$  alors  $\varepsilon_{x_i} = e^{\frac{Y_{x_i} - S^H}{Y_{ref}}} > 1$
- si  $Y_{x_i} < S^B$  alors  $\varepsilon_{x_i} = e^{\frac{S^B - Y_{x_i}}{Y_{ref}}} > 1$

Il faut noter que si 1 point sur 10 se retrouve hors spécifications, il aura effectivement un poids beaucoup plus important que les autres mais cette information risque toutefois d'être trop tempérée par ces derniers pour être prise en compte. On peut donc imaginer un coefficient multiplicateur devant l'exponentielle qui serait fonction du nombre de points considérés suite à la simulation.

### 3.5.3 ALGORITHME DE L'AJUSTEMENT DE LA VALEUR MOYENNE D'UNE PERFORMANCE

L'algorithme est semblable à celui de la réduction de variance. La seule différence est que la fonction à optimiser n'est plus la variance à réduire, mais l'erreur image du mauvais ajustement de la valeur moyenne de la performance vis-à-vis de la cible.

#### a) Pas d'incréméntation

Il reste le même que dans l'étape précédente, et pourra être amélioré de la même manière.

#### b) Diminuer le nombre de facteurs

Comme précédemment, nous utiliserons des routines permettant de limiter le nombre de facteurs. Si ces derniers n'apportent pas d'amélioration intéressante et vérifient donc que  $|\varepsilon_2 - \varepsilon_1| < Pe_1 \max[\varepsilon_2, \varepsilon_1]$ , ils seront fixés. De même, si les facteurs sont très nombreux, on peut fixer les  $(Pe_2 \cdot \text{nb de facteurs})$  facteurs les moins significatifs d'après  $|\varepsilon_2 - \varepsilon_1|$ .

Lorsque l'on est proche d'un minimum local ou que l'on arrive aux bornes de l'intervalle d'intérêt d'un facteur de dessin, on procède comme pour le cas de la réduction de variance.

#### c) Critères d'arrêt

Encore une fois, on s'arrête lorsque le nombre maximum d'itérations est atteint (boucleBmax) ou que tous les facteurs sont fixés. Le paramètre boucleBmax sera supérieur à boucleAmax, pour pouvoir revenir au minimum au point initial si les intérêts de la réduction

de variance et de l'ajustement de  $\mu$  sont totalement divergents. Typiquement, nous prendrons  $\text{boucleBmax}=2.\text{boucleAmax}$ .

### 3.5.4 LES DIFFERENCES ENTRE LES DEUX ETAPES

Hormis que les paramètres  $Pe_1$ ,  $Pe_2$  et  $\text{boucleBmax}$  n'ont pas forcément les mêmes valeurs que leurs homologues de la première étape, il existe une différence de poids entre ces deux algorithmes. Pour que la seconde étape ne se fasse au détriment du gain de la première, il ne faut pas utiliser les facteurs qui dégraderaient la variance; mais c'est prendre le risque de rester très éloigné de la valeur moyenne cible de la performance. Nous pourrions donc réduire l'erreur grâce aux facteurs ayant non pas un  $\Delta\varepsilon$  important, mais un

bon rapport signal sur bruit  $SON = \frac{\frac{\partial\mu}{\partial x_i}}{\frac{\partial Var_y}{\partial x_i}}$ . Ce rapport est toutefois beaucoup moins évident

à manier que  $\Delta\varepsilon$ . Aussi, bien qu'il représente l'un des principaux intérêts de notre approche, nous l'affinerons plus tard, car il semble que dans un premier temps, le gain de la première étape ne soit pas entièrement perdu. Cela s'explique par le fait que les facteurs qui sont conservés le plus longtemps dans le processus de la seconde étape ne sont pas les mêmes que dans la première étape. Les principaux facteurs jouant sur la réduction de la variance  $y$  sont donc fixés plus tôt, ce qui limite l'augmentation de la variance, et la laisse finalement inférieure aux conditions initiales. L'utilisation d'un ratio signal sur bruit devrait améliorer cette première approche, il faut toutefois un certain nombre de facteurs pour pouvoir s'en rendre compte, et de plus qu'ils se comportent différemment : que certains ne jouent que sur  $\mu$ , d'autres que sur  $\sigma$  et d'autres sur les deux...

### 3.5.5 ORGANIGRAMME DE L'AJUSTEMENT DE LA VALEUR MOYENNE DE LA REPONSE

L'organigramme [Figure 3-14] est très semblable à celui de la réduction de variance, si ce n'est la fonction traitée. Seule différence : l'apparition d'un rapport signal sur bruit qui devrait jouer un rôle important dans les problèmes comportant de nombreux facteurs de dessin (plus de 10).



seront des plans de faible résolution (III). Pour moins de 16 facteurs, nous choisirons des plans complets ou des plans de résolutions IV ou V, donnés dans le Tableau 3-2.

| Nombre de facteurs     | Résolution du plan |                                |  |
|------------------------|--------------------|--------------------------------|--|
| 1                      | plan complet       |                                |  |
| 2                      | plan complet       |                                |  |
| 3                      | plan complet       | Tables orthogonales de Taguchi | Colonnes à utiliser                        |
| 4                      | IV                 | L8                             | 1, 2, 4, 7                                 |
| 5                      | V                  | L16                            | 1, 2, 4, 8, 15                             |
| 6, 7, 8                | IV                 | L16                            | 1, 2, 4, 7, 8, 11, 13, 14                  |
| 9,10,11,12,13,14,15,16 | IV                 | L32                            | 1,3,5,7,9,11,13,15,17,19,21,23,25,27,29,31 |

**Tableau 3-2 : Choix des plans d'expériences en fonction du nombre de facteurs**

Le but est d'utiliser les plans de faible résolution pour réduire sensiblement la dimension du problème (nombre de facteurs) à peu de frais, c'est à dire peu de simulations. En dessous de 16 facteurs, on considère des plans de plus forte résolution car leur coût n'est plus prohibitif.

### **3.7 OPTIMISATION DE PLUSIEURS PERFORMANCES SIMULTANEMENT**

Comme nous l'avons précisé, il est important de traiter sérieusement le problème de la fonction coût lorsque l'on a affaire à une seule performance. Cela devient encore plus important dans le cadre d'une optimisation multi-objectifs, mais nous garderons pour le moment la fonction coût définie précédemment.

Nous associons à chaque performance ciblée  $I$  un poids  $P_i$  (phase réduction de la variance) et  $P_i'$  (phase d'ajustement de la valeur moyenne). Si l'on considère les performances  $i$  lors de la seconde phase, nous calculons l'erreur globale associée à une combinaison du plan principal par :  $\varepsilon = \sum_i P_i' \varepsilon_i$ . Les  $\varepsilon_i$  sont calculés d'après les résultats de simulations.

Il reste une voie intéressante à explorer qu'il faudrait comparer à cette approche. Dans le cas où une estimation du rendement est réalisée avant l'optimisation, et c'est souvent le cas pour pouvoir juger de l'amélioration apportée, on peut envisager de se servir de ces résultats initiaux. Ainsi, l'erreur globale associée à une combinaison du plan principal

s'écrirait :  $\varepsilon_{N_1} = \sum_{i=1}^{perf\ n} P_i' \cdot \frac{\varepsilon_{N_1}^i}{\varepsilon_{initial}^i}$ . Cette approche présente l'avantage d'éviter la division par

Yref lors du calcul de la fonction erreur, dangereuse, qui n'est plus nécessaire. Son inconvénient est qu'une estimation du rendement est théoriquement nécessaire après la réalisation de chaque plan produit. Se baser sur les quelques simulations réalisées lors du plan produit pour estimer le rendement est trop hasardeux, et réaliser une estimation par une analyse de Monte-Carlo classique trop coûteux en temps, mais cette approche serait certainement très intéressante si une méthode rapide d'estimation du rendement était implémentée (§2.3) et que l'on avait affaire à des MEMS peu complexes et simulables rapidement au sein d'une netlist. Elle donnerait a priori de meilleurs résultats dans ces conditions, et pourrait être proposée comme une option quand les performances ciblées sont très nombreuses et que le MEMS est très simple.

Dans ce chapitre, nous avons détaillé l'algorithme mis au point pour réaliser l'optimisation statistique du rendement paramétrique de microsystèmes. Il fait appel à la méthode des plans d'expériences et comporte deux phases principales : la réduction de variance des performances puis l'ajustement de ces performances. Cette approche est choisie car elle permet une diminution sensible de la variance lorsque cela est possible, contrairement aux méthodes habituelles dans lesquelles la variance est une donnée de second ordre, quand elle est considérée.

# Chapitre 4

## IMPLEMENTATION DE L'ALGORITHME, APPLICATIONS ET RESULTATS



## 4. IMPLEMENTATION DE L'ALGORITHME, APPLICATIONS ET RESULTATS

Nous abordons dans ce chapitre le problème de la modélisation des MEMS, essentiellement du à leur aspect multi-physique, aspect que nous retrouvons dans les simulateurs utilisés dans ce domaine. Puis nous présenterons la structure générale du logiciel développé pendant la seconde moitié de la thèse et aborderons ses points sensibles. Enfin, nous le mettrons en œuvre en considérant deux exemples : celui d'un filtre RLC passe-bande et celui d'un microrésonateur MEMS.

### **4.1 MODELISATION DES MEMS**

Il n'existe pas de langage spécifique aux MEMS. Suivant les cas, on utilise des modèles éléments finis, des modèles comportementaux type Verilog-A ou des modèles équivalents de type SPICE.

#### 4.1.1 DIFFICULTES DE MODELISATION ET DE CONCEPTION

Les paramètres de dessin et les paramètres technologiques doivent clairement apparaître dans les modèles et dans leurs instances situées dans la netlist, afin d'être accessibles à l'optimiseur. Nous devons donc traiter des modèles paramétrés, c'est le point commun à tous les modèles. Mais il se pose ensuite d'autres problèmes plus spécifiques : les phénomènes physiques à cette échelle sont trop complexes pour être modélisés trop sommairement [BONI99].

#### **a) Simulations multi-domaines**

Les actionneurs les plus courants sont les peignes électrostatiques, utilisés notamment dans les accéléromètres. Pour pouvoir les simuler, il faut connaître certaines de leurs caractéristiques, notamment des capacités qui ne sont pas simples à modéliser et qui le sont donc généralement par le biais de l'utilisation de la méthode des éléments finis (FEM) ou des éléments aux frontières (BEM) [WHIT99]. Dans le premier cas, c'est l'ensemble de l'espace qui est "maillé" pour permettre un calcul de proche en proche; dans le second, on ne s'intéresse qu'à la surface des volumes. Mais les peignes électrostatiques comportent aussi une partie mécanique non moins importante, bien que plus aisée à modéliser. La meilleure solution pour connaître la réponse d'un tel système est d'utiliser un simulateur multi-physique

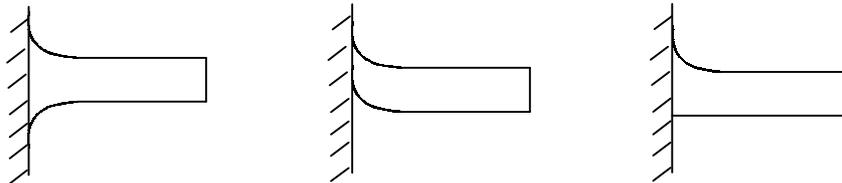
(ou la co-simulation), qui traitera en même temps l'aspect mécanique et électrostatique du problème. Ces simulateurs sont très lents : ils font appel à des méthodes type FEM ou BEM, et nécessitent un nombre très important de calculs qui les rend peu attractifs dans le cadre d'une CAO "simple" dédiée aux MEMS. On peut essayer de construire, à partir des réponses de systèmes obtenues par les éléments finis, des modèles comportementaux plus légers qui seront implantés dans une netlist et rapidement simulés. MEMSModeler, développé par la société MEMSCAP, permet d'obtenir de tels modèles. Sans cela, la plupart des modèles de MEMS sont donc assez rudimentaires ou ont un domaine de validité restreint.

### **b) Importance de l'usure**

Les MEMS comportent des structures mobiles : toutes les formes de détérioration de structures mécaniques devraient donc être prises en compte afin de pouvoir établir un modèle de vieillissement pour chaque MEMS. De plus, il existe à l'échelle microscopique des phénomènes non visibles en mécanique classique, mais que l'on ne peut pas négliger ici. Ces modèles sont intimement liés à la technologie, et ici encore, une communication importante entre concepteurs et technologues est de mise.

### **c) Problèmes pour une technologie donnée**

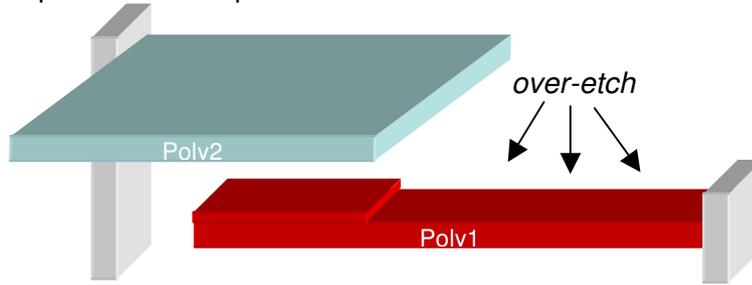
On peut facilement modéliser le comportement d'une simple poutre ayant une extrémité libre, et l'autre fixe. Mais suivant la technologie employée, l'encastrement de la poutre peut prendre différents aspects [Figure 4-1], et donc les déplacements de cette dernière sous l'effort ne se feront pas toujours comme prévu si l'on utilise un modèle trop rudimentaire. Ainsi, certaines approximations (capacité plane...) peuvent s'avérer très dangereuses. Une bonne modélisation doit donc tenir compte de nombreux paramètres liés au processus de fabrication et n'est toujours pas simple à obtenir.



**Figure 4-1 : La forme d'une simple poutre est fonction de la technologie utilisée**

Par ailleurs, les étapes suivantes du processus de fabrication sont aussi cruciales : cette poutre n'aura pas la même forme, notamment la même épaisseur, si elle est très exposée ou à l'inverse, si elle se situe sous une structure plus large [Figure 4-2]. Il faut donc modéliser la poutre en fonction de la technologie utilisée, mais aussi des éventuelles structures environnantes. On peut limiter ces cas complexes, pour lesquels on ne dispose

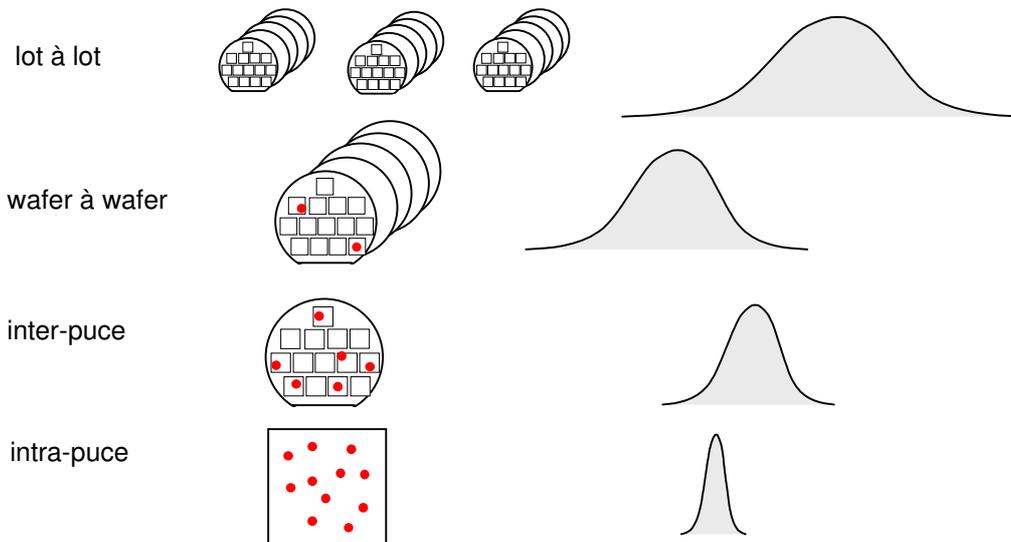
pas forcément de modèles appropriés tant ils sont nombreux, mais cela limite malheureusement le potentiel offert par les structures en 3D.



**Figure 4-2 : La partie non protégée de la poutre de droite subit une sur-gravure**

#### d) Le device mismatch

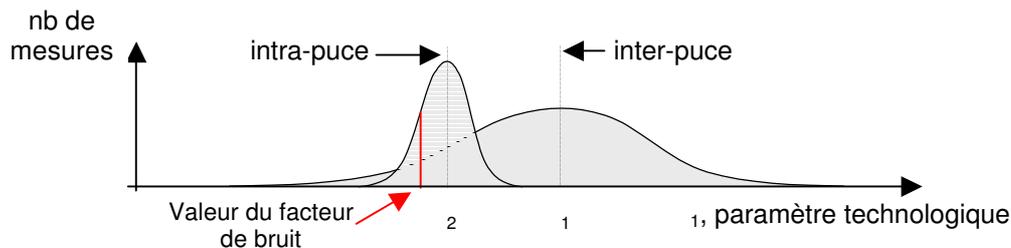
C'est un phénomène auquel on est confronté lorsque deux structures identiques sont utilisées dans un système, mais qu'en raison des variations de la technologie au sein de la puce, la symétrie entre les deux structures n'est pas exactement assurée [BONI99]. Cela peut entraîner un déséquilibre en fonctionnement, il faut donc pouvoir en tenir compte lors de la phase de conception. Les variations de la technologie sont suivies à différents niveaux [Figure 4-3].



**Figure 4-3 : Dispersion des paramètres technologiques**

Ces variations deviennent de plus en plus critiques alors que les microtechnologies bénéficient de gravures de plus en plus fines. Elles font l'objet de nombreuses recherches [STIN97] [OUMA96] [STIN96] [CHAN95] [BONI94]. Les technologues sont intéressés par toutes ces nuances, mais pour les concepteurs qui sont supposés disposer d'une technologie assez stable, on se contente de considérer les variations inter voire intra-puce dans le cadre de l'optimisation statistique du rendement [MEHR00] [BONI96]. Si le concepteur désire connaître les effets d'un device mismatch qui va dégrader les

performances de son MEMS et qu'il peut se permettre quelques milliers de simulations, il utilisera l'optimiseur qui fixera les facteurs de bruit de manière particulière à chaque simulation. Au lieu de réaliser un seul tirage aléatoire sur une fonction ayant une distribution de type normale, centrée en  $\mu$  et  $\sigma$  écart type  $\sigma$ , on réalisera deux : un premier sur la distribution liée à l'aspect inter-puce qui fixera la valeur moyenne pour le second tirage effectué sur la distribution liée à la dispersion intra-puce [Figure 4-4].



**Figure 4-4 : Choix de la valeur d'un facteur de bruit**

Des méthodes sont proposées dans la littérature pour tenter de prévoir ces dispersions [CARL97]. Le NMRC (National Microelectronics Research Center-Irlande) [NMRC] a notamment consacré beaucoup d'efforts à la caractérisation statistique de technologies CMOS [DIAZ97] [MCCA98] [HEAL99].

#### 4.1.2 MODELES SPICE

En CAO microélectronique, les simulateurs traitent des netlists où les circuits sont décrits sous forme d'assemblage de modèles de composants dont on précise la valeur des paramètres. Le langage de description le plus utilisé est SPICE. Décrire des MEMS dans un tel langage revient à essayer de traduire les équations régissant le comportement des MEMS sous la forme d'un circuit électronique dont la mise en équations est identique. Par ailleurs, les équations utilisées pour décrire ces comportements sont souvent des approximations : module d'Young équivalent pour des structures composées de plusieurs couches, approximation de capacités planes lors de calculs de pression électrostatiques sur des peignes ou des électrodes... Le résultat final donne généralement une image assez grossière de la réalité.

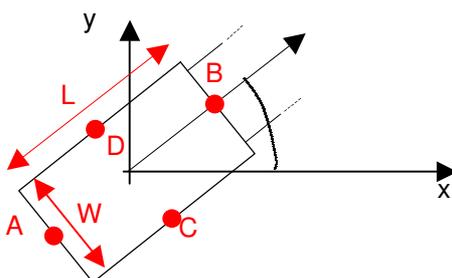
#### 4.1.3 MODELES COMPORTEMENTAUX

Les modèles comportementaux sont pour le moment les plus intéressants pour la CAO MEMS [SWAR98]. Ce sont des boîtes noires comportant des modèles analytiques. Ces derniers retranscrivent directement les équations décrivant le comportement des structures MEMS [GABB98]. Comme précédemment, la justesse de ces équations est critique, mais

l'avantage est de n'avoir aucune approximation ou traduction de ces dernières avant de passer à la phase de simulation.

Les outils Cadence et Mentor Graphics restent les principaux environnements de conception en microélectronique analogique, et donc pour le moment pour les MEMS. Pour Cadence, c'est le Verilog-A qui est utilisé comme langage de description, alors que VHDL-AMS [HANN99] est utilisé par Mentor en remplacement du HDLA avec le simulateur ADVance MS. Il suit la norme IEEE VHDL-AMS, comme le Verilog-A.

Pour pouvoir analyser l'influence des incertitudes des paramètres technologiques sur les performances du MEMS, il faut pouvoir accéder à ces derniers de manière très "précise", et nous devons définir une approche pour les MEMS. Considérons un composant élémentaire : une plaque à quatre cotés. Son modèle devra avoir certaines particularités, notamment celle d'être orienté par rapport aux autres composants du système [Figure 4-5]. Dans ce type de modèle, on établit des connexions avec d'autres composants en stipulant quelles faces sont en contact en quel point.



**Figure 4-5 : Une plaque : composant élémentaire orienté**

On peut faire apparaître dans la netlist les incertitudes existantes pour chaque couche de la technologie. Par exemple, si l'on considère la couche un de polysilicium ( $P_1$ ), nous mettrons dans la netlist les paramètres  $P_1\delta_x = 0, P_1\delta_y = 0$  (si l'on raisonne en 2D) qui correspondent à l'incertitude sur les dimensions de la couche suivant l'axe des x et des y et peuvent être interprétés comme un bruit "intégré". Si la plaque de la Figure 4-5 apparaît dans cette netlist, ses dimensions figureront dans son modèle sous la forme :

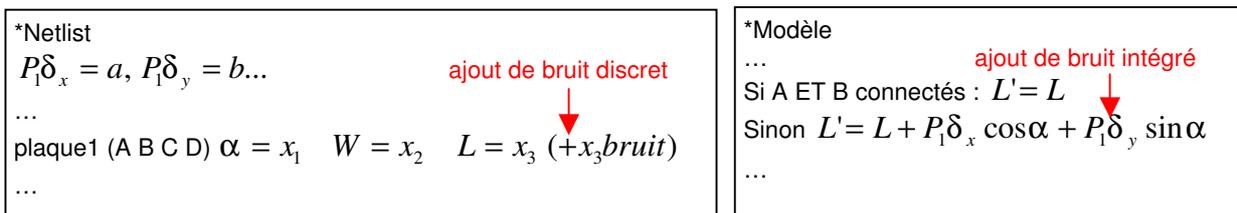
si A ET B sont connectés :  $L' = L$ ,  $L'$  étant la valeur de L dans le modèle.

sinon :  $L' = L + P_1\delta_x \cos\alpha + P_1\delta_y \sin\alpha$ .

Ainsi, lorsque l'on désirera réaliser une optimisation statistique du rendement, nous choisirons dans la netlist  $P_1\delta_x$  et  $P_1\delta_y$  comme paramètres de bruit "intégrés". Ce genre de paramètre qui s'applique à tous les composants de la netlist, comme le module d'Young ou

la résistivité des couches, n'est parfois pas donné directement dans la netlist mais dans un fichier "technofile" pour plus de lisibilité.

Si l'on se limite à ce type de modélisation, nous ne pourrons pas tenir compte de l'effet de device mismatch, car c'est la même variation qui est appliquée ici à toutes les structures réalisées avec  $P_1$ . Il faut donc que le concepteur considère un bruit supplémentaire dont l'effet s'additionnera au précédent. C'est pourquoi chaque dimension géométrique qui est passée en paramètre dans la netlist peut en plus se voir parasitée par l'ajout d'un paramètre de bruit "discret". Le modèle n'est pas à modifier pour tenir compte de cette étape, la valeur du paramètre dans ce cas là est ajustée lors de l'écriture de la netlist [Figure 4-6].



**Figure 4-6 : Netlist et modèle comportant des paramètres de bruit *intégrés* et *discrets***

On peut déterminer ainsi quelques composants de bases tels que les plaques, les poutres ou les ancrages [BAIDYa99] [BAIDYb99]. Mais définir et utiliser des composants plus complexes qui ne seraient pas composés de ces éléments de bases devient compliqué pour un concepteur. Ce dernier peut donc par exemple utiliser un outil utilisant la méthode des éléments finis qui lui fournira ensuite un jeu d'équations réduites lui permettant de connaître la réponse du MEMS, par exemple un déplacement, aux points qui l'intéresse. Ce modèle permet de construire un modèle comportemental paramétré, rapide à simuler.

#### 4.1.4 MODELES FEM

Les modèles obtenus par le biais de la méthode des éléments finis sont très proches du comportement réel des structures MEMS. L'inconvénient est le coût de calcul par une telle méthode qui est prohibitif pour une CAO qui se veut simple est rapide pour concevoir des systèmes ou évaluer des technologies [WILS99]. On peut par contre imaginer une librairie de modèles comportementaux réalisés sur la base des systèmes de matrices définies par la méthode des éléments finis. Au cas où un concepteur ne trouve pas les composants (et donc leur modèle) nécessaires à la réalisation de sa structure dans une librairie, il doit pouvoir utiliser un outil permettant de faire de l'analyse éléments finis (FEA) et d'en déduire les équations intéressantes.

#### 4.1.5 CHOIX DE NOS MODELES

Les modèles que nous utilisons dans le cadre de notre étude sont des modèles SPICE pour les composants électroniques, et des modèles Verilog-A pour les composants MEMS. Les circuits sont simulés avec Spectre, le simulateur de Cadence. Nous n'avons pas encore intégré à nos modèles les paramètres de type  $P_1\delta_x$  et  $P_1\delta_y$  permettant de traiter les paramètres de bruit intégrés, nous réaliserons donc notre optimisation du rendement en considérant seulement les paramètres de bruit discrets, ce qui est a priori suffisant pour valider notre approche.

### **4.2 CONSIDERATION DE L'INFORMATIQUE**

L'approche choisie pour l'optimisation statistique du rendement ayant été définie et justifiée, il s'agit maintenant de l'implémenter et de mettre au point le logiciel de manière à faciliter les tests d'évaluation pour une version préliminaire. L'algorithme peut faire appel à des milliers de simulations lorsque l'on désire traiter un MEMS en prenant en compte un nombre important de paramètres. Les résultats à traiter sont donc très nombreux. Pour valider l'approche, il faudra aussi la tester sur plusieurs MEMS : on ne peut se contenter d'une netlist figée. L'implémentation s'avère donc assez lourde, même pour une version d'évaluation.

#### 4.2.1 ENVIRONNEMENT

La plupart des concepteurs de circuits microélectroniques complexes travaillent sous stations UNIX, mais Memscap fournit des solutions pour PC (Windows) et stations UNIX. Sous UNIX, le logiciel prend la forme d'un module de Cadence ou Mentor Graphics, utilisant leur simulateur habituel, Spectre et Eldo, qui gèrent très bien les modèles comportementaux et traitent des circuits de très grande taille. Mais les concepteurs de MEMS ne sont pas tous issus de ce milieu et il existe une demande importante pour un environnement de conception sous PC. Pour cette solution, le simulateur impliqué est T-Spice. La prise en charge des modèles comportementaux n'y est pas très aisée, et le simulateur peut peiner lors du traitement de circuit très importants. Cette solution reste néanmoins très appréciée car il n'est pas nécessaire de posséder Cadence ou Mentor Graphics, très chers, pour se lancer dans la conception de MEMS, et ils ne sont pas simulés fréquemment au sein de circuits très complexes.

Pour s'affranchir de l'environnement PC ou UNIX (portabilité), le langage Java semblait bien adapté. Le C++ est réputé encore un peu plus rapide, mais ce sont les simulations qui

occupent 99% du temps mis à réaliser l'optimisation, ce critère n'entre donc pas en jeu. Java permettant de plus un développement et une maintenance très aisée du code, c'est ce langage orienté objet que nous avons choisi.

Concernant le choix du simulateur, le marché de la conception microélectronique analogique étant dominé par Cadence, c'est sur son simulateur Spectre que notre choix s'est porté. Les modèles comportementaux considérés lors de l'optimisation seront donc écrits en Verilog-A.

#### 4.2.2 STRUCTURE IMPLEMENTEE

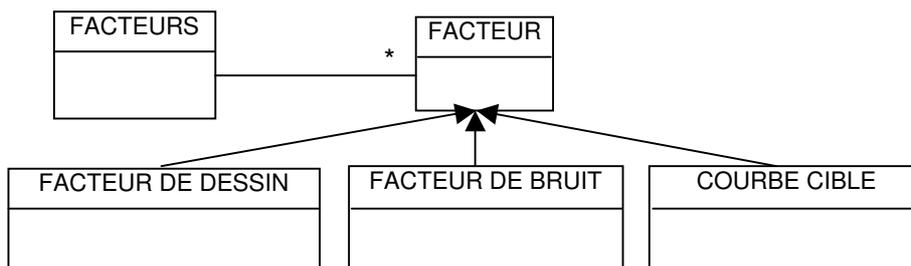
Elle est fonction de l'algorithme, du choix des objets et des fichiers que l'on désire manipuler en cours d'optimisation. Un soin particulier (optimisation du code) doit être apporté à la réécriture de la netlist et à la lecture des résultats de simulation, cette opération pouvant être réalisée de très nombreuses fois suivant les plans produits [Figure 4-7]. Ce chapitre donne les principales classes utilisées, ainsi que les fichiers utilisés par le programme.



**Figure 4-7 : La réécriture de la netlist doit être simple et rapide**

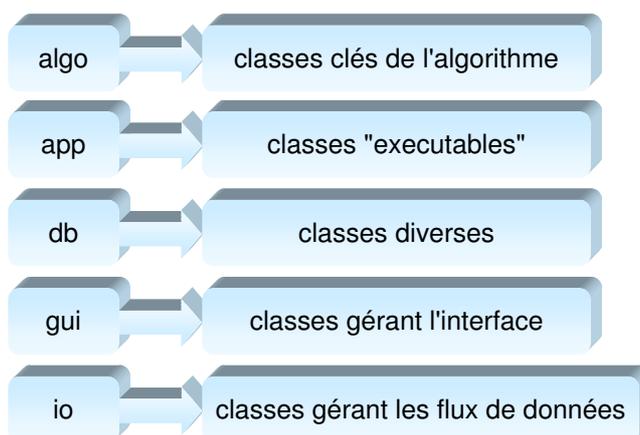
##### **a) Choix des classes**

Il s'agit tout d'abord de considérer les classes à la base du problème : les paramètres de dessin et de bruit (discrets), et les paramètres technologiques qui sont des paramètres de bruit intégrés ainsi que les performances ciblées appartiennent tous à des classes distinctes, ayant de nombreux points communs. Ils partagent donc plusieurs méthodes identiques, notamment des facilités de lecture-écriture, appliquées aux variables de ces objets, de type String, Boolean ou Double [Figure 4-8].



**Figure 4-8 : Diagramme de classes des données de base**

La seconde série de classes à définir concerne l'algorithme, une classe sera donc dédiée à l'estimation du rendement, une autre à la réduction de la variation, et une dernière à l'ajustement de la valeur moyenne. Bien qu'il constitue une part importante du temps de travail de cette thèse, décrire en détail l'ensemble du code (classes et méthodes) serait fastidieux et peu intéressant : l'implémentation compte plus de 4000 lignes de code. Nous nous contenterons de reprendre les grandes lignes en annexe [p. 153]. Elles sont rangées dans 5 répertoires suivant leur fonction [Figure 4-9].



**Figure 4-9 : Les classes sont rangées par fonction**

### **b) Les paramètres mixtes : un point sensible**

La partie sensible du code concerne les paramètres mixtes : ils sont à la fois des facteurs de dessin et des facteurs de bruit. Ainsi, lorsque l'on réalise un plan produit et que l'on considère une expérience du plan principal, les facteurs de dessin sont fixés. Si l'on considère l'un de ces paramètres qui s'avère être mixte, sa valeur dans la netlist sera pourtant modifiée à chaque nouvelle simulation requise par le plan bruit par l'ajout de bruit. Pour cette raison, deux valeurs sont associées à chaque paramètre : une valeur qui reste fixe pour une combinaison donnée du plan principal et qui est contenue dans la classe `AbstractParameter`, et une valeur courante du paramètre mise éventuellement à jour à chaque réécriture de la netlist, contenue dans la classe `InstanceParameter` (qui descend

d'AbstractParameter). On traitera différemment un paramètre de bruit discret et un paramètre de bruit intégré (paramètre technologique). Les niveaux du plan bruit sont calculés tels que :

niveau bas = moyenne de la dispersion du paramètre - x.écart type de la dispersion

niveau haut = moyenne de la dispersion du paramètre + x.écart type de la dispersion

x est une valeur que l'utilisateur peut fixer à son gré (voir aussi § 4.3).

Si un paramètre de bruit discret est aussi mixte, sa valeur lors d'une simulation sera la valeur imposée par le plan principal plus le niveau bas ou haut imposé par le plan bruit.

S'il s'agit d'un paramètre technologique, on conserve la même formule, mais il est superflu de préciser la valeur moyenne de la dispersion : on considère de toute manière que c'est la valeur donnée dans le *technofile* ou la *netlist*.

### **c) Gestion des fichiers**

De nombreuses données sont saisies dans des fichiers, que ce soit pour la saisie du problème ou le suivi de l'évolution des facteurs en cours d'optimisation. Le détail du contenu et de l'intérêt de ces fichiers est donné en annexe [p.153].

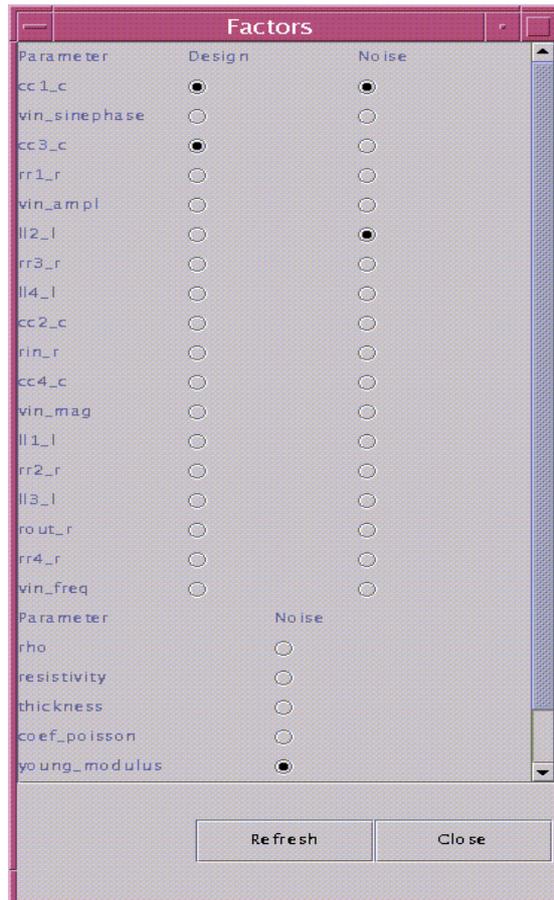
### **d) Appel de programmes extérieurs**

Trois programmes extérieurs sont sollicités : le simulateur Spectre, le traceur Gnuplot et l'afficheur XView. L'appel à Spectre est sommaire : il faut prendre garde à ce qu'il y ait toujours une licence de libre en cours d'optimisation car de nombreuses simulations sont requises, et la licence est à chaque fois libérée. Une protection doit être implémentée à ce niveau-là. De même, l'utilisation de "threads" serait intéressante pour visualiser plusieurs courbes sous différents graphes Gnuplot en même temps. Cela dit, la version actuelle n'est qu'une version de démonstration de validité de l'approche, et fonctionne assez bien moyennant quelques précautions (disponibilités des licences, existence des fichiers recherchés par le programme en phase d'initialisation, etc.).

## **4.3 INTERFACE DU LOGICIEL**

Nous détaillons ici l'interface du logiciel. Sa structure devait initialement être assez rigide pour limiter le temps passé en programmation. Mais pour mettre au point l'algorithme, nous ne pouvions pas nous contenter d'une netlist fixe car de nombreux tests sont nécessaires pour le valider. Nous avons donc écrit un *parser*, qui permet de lire les netlists (comportant éventuellement des modèles Verilog-A) écrites pour le simulateur Spectre, et de les traduire pour être traitées et réécrites par le logiciel. Le choix et l'analyse des résultats de centaines de simulations nécessite aussi un haut degré d'automatisation.

Après la lecture de la netlist et l'identification par le programme de tous les paramètres, une interface [Figure 4-10] propose à l'utilisateur de sélectionner les paramètres de bruit discret, les paramètres de dessin, et les paramètres technologiques (bruit intégré). Ces derniers sont situés dans un autre fichier appelé *technofile*. Dans l'exemple de la figure suivante, cc1\_c est un paramètre mixte, cc3\_c est un paramètre de dessin, ll2-l est un paramètre de bruit discret et le module d'Young est un paramètre de bruit intégré, comme tous les paramètres "technologiques".



**Figure 4-10 : Interface de sélection des paramètres manipulés lors de l'optimisation**

Les paramètres sélectionnés possèdent certains attributs. Pour les paramètres de dessin, ces données sont stockées dans un fichier nommé *D\_nom-de-l'instance\_nom-du-paramètre*. On trouvera ainsi dans le fichier *D\_cc1\_c* les caractéristiques propres au paramètre "c" de l'instance cc1 sous le format "nom min max pas", par exemple :

"cc1\_c 60e-9 100e-9 4e-9".

De même, les attributs des paramètres de bruit sont stockés dans un fichier nommé *N\_nom-de-l'instance\_nom-du-paramètre* pour du bruit discret et *N\_ nom-du-paramètre* pour du bruit intégré, où l'on perd donc la référence à une instance donnée. On trouvera ainsi

dans le fichier N\_cc1\_c les caractéristiques propres au paramètres "c" de l'instance cc1 sous le format "nom -9 0 2". On y trouve bien sûr la dispersion du paramètre de bruit, la moyenne de sa distribution, et . Cette variable permet de calculer les niveaux bas et haut de ce facteur dans le plan bruit donnés par +/- à chaque case cochée dans l'interface [Figure 4-10] correspond un fichier où sont stockées les données des paramètres. D'autres fichiers sont utilisés par l'algorithme mais ne sont pas encore accessibles à l'interface elle-même. Leur contenu est donc "figé", ils sont décrits dans les exemples du chapitre suivant.

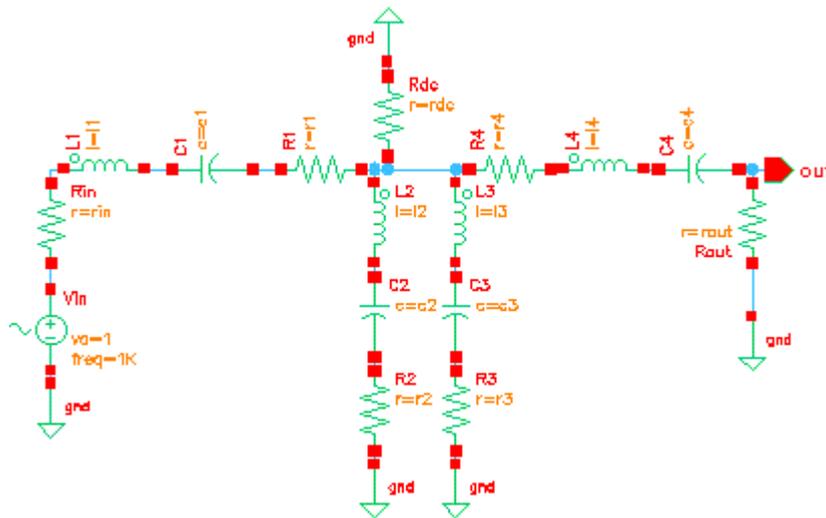
#### **4.4 EXEMPLE : FILTRE PASSE BANDE RLC**

Nous désirons améliorer le rendement d'un filtre passe-bande RLC. La performance qui nous intéresse est sa réponse fréquentielle, mais une analyse de Monte-Carlo nous prédit un mauvais rendement. Nous désirons ici travailler plus sur l'amélioration du rendement que sur la réduction de la variabilité. Nous ne concéderons donc que peu d'itérations à cette dernière. Par ailleurs, pour finaliser le processus d'optimisation, nous utiliserons à nouveau la deuxième étape (réduction de l'erreur) en considérant un pas beaucoup plus faible pour l'ensemble des facteurs de dessin. Il est effectivement possible que certains de ces facteurs aient été écartés du processus d'optimisation dès les premières itérations, mais qu'ils retrouvent un certain intérêt en phase finale. Il peut être important de réévaluer leur influence à proximité de la solution choisie.

##### 4.4.1 SAISIE DU PROBLEME

###### **a) Topologie du circuit**

La netlist peut être créée directement ou par l'intermédiaire d'un éditeur de schéma, par exemple Virtuoso (Cadence) [Figure 4-11]. Le type d'analyse n'est pas défini à ce stade.



**Figure 4-11 : Circuit RLC à optimiser**

Sur cette figure, les paramètres sont associés à des variables. Cela n'est pas le cas dans notre netlist où tous les paramètres sont identifiés par le *parser* (lecteur de netlist) sans avoir recours à des mots clés comme *parameter*. La netlist correspondant à ce circuit est donc donnée avec les valeurs initiales des paramètres, elle est mise en forme par le logiciel pour faciliter ses nombreux traitements ultérieurs :

```
*Netlist written for otpimization
global gnd
simulator lang=spectre
vin (4 gnd) vsource type=sine fundname="Fname9" ampl=1.0 freq=0.16 sinephase=0.0 mag=1.0
rin (4 1) resistor r=1200.0
ll1 (2 1) inductor l=1.824
cc1 (2 3) capacitor c=9.443999999999999E-8
rr1 (3 5) resistor r=287.0
ll2 (6 5) inductor l=0.4272
cc2 (6 7) capacitor c=9.88E-7
rr2 (7 gnd) resistor r=66.0
ll3 (8 5) inductor l=0.1437
cc3 (8 9) capacitor c=3.4E-7
rr3 (9 gnd) resistor r=22.5
ll4 (11 10) inductor l=1.824
cc4 (11 out) capacitor c=7.87E-8
rr4 (5 10) resistor r=287.0
out (out gnd) resistor r=1200.0
rdc (5 gnd) resistor r=0.1Meg
```

**b) Choix des paramètres et de leurs caractéristiques**

Cette étape a été détaillée dans le chapitre précédent à la page 120. Les paramètres de différentes natures sont sélectionnés dans une interfaces, créée d'après la lecture de la netlist à traiter, et leurs caractéristiques sont entrés dans des fichiers portant leurs noms.

### c) Choix des paramètres de l'algorithme

Les paramètres  $Ps_1$ ,  $Ps_2$ ,  $Ps_3$ ,  $Pe_1$  et  $Pe_2$  décrits aux pages 97 et 104 sont donnés dans cet ordre dans le fichier *algodata*. On y trouve aussi le nombre maximum d'itérations toléré pour la première et la deuxième phase de l'algorithme (*loopAmax* et *loopBmax*), ainsi que pour l'ajustement éventuel qui reprend tous les facteurs avec un pas réduit sur quelques itérations en fin de processus (*loopB2max*). Les valeurs, pour notre exemple, seront :

$Ps_1=0.001\%$      $Ps_2=0$      $Ps_3=0.001\%$      $Pe_1=0.001\%$      $Pe_2=0$     *loopAmax*=4  
*loopBmax*=5    *loopB2max*=3.

é par deux dans la phase finale.

La principale faiblesse du logiciel réside dans ces données : elles sont pour le moment figées alors que pour un fonctionnement optimal, il faudrait les évaluer en fonction du circuit à traiter.

### d) Choix des spécifications

Les caractéristiques de chaque spécifications sont stockées dans des fichiers nommés : *T\_type-d'analyse\_nœud\_nom-de-la-netlist*. Le format de la première ligne est :

*nom du fichier*    *type de l'analyse*    *nœud où la tension est prise*    *poids de cette spécification*  
*début de la simulation*    *fin de la simulation*    *pas pour l'affichage*    *affichage linéaire ou*  
*logarithmique.*

Ainsi, pour une analyse fréquentielle de la netlist *passivebpf* de 200 à 800 Hz, avec 100 points lors de l'affichage logarithmique de l'expression  $20 \log(v(\text{out}))$ , nous écrirons :

*ac\_out\_passivebpf ac out 1 200 800 100 db*

Les spécifications elles-mêmes sont notées dans la suite du fichier sous la forme :

*fréquence*    *valeur cible*    *valeur minimum*    *valeur maximum*

Cette ligne est générée automatiquement par un programme annexe qui calcule la fréquence en fonction de l'affichage désiré : 100 points correspond à 100 fréquences équitablement réparties suivant l'affichage. Le programme remplace les valeurs par des NaN (*Not a Number*). Par la suite, l'utilisateur n'a plus qu'à remplacer ces NaN par des valeurs, là où il en a l'utilité. Dans notre exemple, nous fixons des valeurs maximales et minimales à ne pas dépasser en 6 fréquences que l'on juge importantes : 210, 240, 360, 490, 700 et 770 Hz. Pour ces points, nous modifierons donc les lignes correspondantes :

*ac\_out\_passivebpf2 ac out 1 200 800 100 db*  
199.9999999999983    NaN    NaN    NaN  
202.79189595800554    NaN    NaN    NaN  
205.62276533121315    NaN    NaN    NaN

```

208.49315216822401  NaN  NaN  NaN
211.4036081122759  NaN  -52  -46
214.35469250725836  NaN  NaN  NaN
...
239.49574092378546  NaN  -52  -46
242.83897687900924  NaN  NaN  NaN
...
353.0811985162618  NaN  NaN  NaN
358.01002837118864  NaN  -16  -10
363.0076621268642  NaN  NaN  NaN
...
485.67795375801853  NaN  NaN  NaN
492.45776533796607  NaN  -16  -10
499.3322195606446  NaN  NaN  NaN
...
686.8523491502028  NaN  NaN  NaN
696.4404506368987  NaN  -65  -40
706.1623970325229  NaN  NaN  NaN
...
756.8461173804762  NaN  NaN  NaN
767.4112954602106  NaN  -65  -40
...

```

Ces spécifications sont tracées, ainsi que les courbes correspondant aux simulations. Cet affichage n'est pas superflu, il permet notamment de détecter certaines dérives intrinsèques à l'algorithme qui sont moins faciles à interpréter si l'on doit se contenter d'analyser les chiffres.

#### 4.4.2 SUIVI DU PROCESSUS D'OPTIMISATION

Il est important dans la phase de mise au point du programme de pouvoir suivre l'évolution des valeurs de toutes les variables du problème, ainsi que le bon déroulement des phases successives du processus d'optimisation. De même, chaque fois qu'un facteur est fixé, la raison en est donnée de manière à vérifier que cela ne vient pas d'un mauvais réglage des paramètres de l'algorithme. Pour une analyse portant sur un seul facteur, on trouve le suivi des modifications qui lui ont été apportées dans un fichier nommé *DV\_nom-de-l'instance\_nom-du-paramètre*. Avec les spécifications données précédemment et en considérant toutes les capacités et les inductances comme des paramètres mixtes, le fichier DV\_cc4\_c prend la forme suivante [Figure 4-12] :

```

initial value=7.87E-8  MIN=6.0E-8  MAX=1.0E-7  STEP=4.0E-9
All values have been updated by FactorCretor.setFactorsAttributes: new values are MIN=6.296E-8
MAX=9.443999999999999E-8  STEP=5.9025E-10
Sigma reduction
LoopA=1
L1=7.8404875E-8      L2=7.899512499999999E-8
E1=0.005442952097250449  E2=0.005841562161025124  deltaE=3.9861006377467454E-4
S1=1.4694432350983542    S2=1.468911065442388      deltaS=-5.321696559661415E-4
DF is fixed at EpsMin: epsilon is degraded beyond
|epsilon2-epsilon1|>Ps3*Max(epsilon2,epsilon1)/100=5.841562161025123E-8

```

**NoptA=7.8404875E-8**  
**Mean Tuning**  
**LoopB=1**  
L1=7.810975000000001E-8      L2=7.87E-8  
E1=0.002254386272569298      E2=0.002024472849864066      deltaE=-2.2991342270523192E-4  
S1=1.4191299970949773      S2=1.4178319300506967      deltaS=-0.001298067044280593  
**LoopB=2**  
L1=7.929025000000001E-8      L2=7.98805E-8  
E1=8.488016405346452E-4      E2=0.001160085702781781      deltaE=3.1128406224713583E-4  
S1=1.4327568887920916      S2=1.4316173714614338      deltaS=-0.0011395173306578688  
**LoopB=3**  
L1=7.889675E-8      L2=7.909349999999999E-8  
E1=6.937925591456682E-4      E2=6.944681009636967E-4      deltaE=6.755418180284887E-7  
S1=1.4354070185485126      S2=1.4349533366027896      deltaS=-4.53681945723039E-4  
DF is fixed: stopstep was=true  
**NoptB=7.889675E-8**  
**Mean Tuning** : all DF are considered a last time with loopBMax=3  
**LoopB=1**  
L1=7.87491875E-8      L2=7.90443125E-8  
E1=6.804056926877685E-4      E2=6.812289295430465E-4      deltaE=8.232368552779602E-7  
S1=1.4353535117771559      S2=1.4345078956378472      deltaS=-8.456161393086692E-4  
**LoopB=2**  
L1=7.81589375E-8      L2=7.84540625E-8  
E1=6.809107777314503E-4      E2=6.803834900989177E-4      deltaE=-5.272876325325415E-7  
S1=1.4327324202204694      S2=1.431773023079752      deltaS=-9.593971407173552E-4  
**LoopB=3**  
L1=7.85524375E-8      L2=7.86508125E-8  
E1=5.8955976964782586E-5      E2=5.901661943121333E-5      deltaE=6.064246643074631E-8  
S1=1.4336422168763059      S2=1.4333394558425856      deltaS=-3.0276103372028906E-4  
DF is fixed: stopstep was=true  
**NoptB=7.85524375E-8**

**Figure 4-12 : Fichier de suivi de l'évolution des valeurs d'un facteur de dessin**

Au début du fichier, on note que les valeurs saisies dans le fichier de description de cc4\_c sont remplacées. Cette substitution s'effectue par une partie du programme, que l'on peut décider d'ignorer, et qui permet de fixer d'un seul coup toutes les bornes MIN et MAX et le pas de l'ensemble des paramètres. Cela se fait au choix en pourcentage de la valeur initiale ou en nombre d'écart type lorsqu'il s'agit de paramètres déclarés comme étant mixtes. L1 et L2 sont les niveaux bas et haut des paramètres du plan principal. Lors de la phase de réduction de la variance, le paramètre cc4\_c est fixé rapidement car l'utiliser pour réduire la variance dégraderait trop la réponse moyenne de la performance. Modifier le paramètre d'algorithme Ps<sub>3</sub> permettrait éventuellement de conserver cc4\_c pour quelques itérations supplémentaires. Lors de la seconde phase de l'algorithme (*mean tuning*), une tendance est estimée lors de la première itération : bas et haut pour la prochaine itération [Figure 3-7 : Pas d'incrémentation]. Mais cela mène par la suite à ôt dans l'autre direction. On se retrouve dans le cas où l'on semble être passer par un minimum de la fonction d'erreur. Lors de la prochaine itération, on essaiera de l'approcher plus précisément, puis ce facteur sera

fixé [Figure 3-9 : Oscillations autour d'un minimum]. *stopstep* est le booléen correspondant à FixFact dans l'organigramme de la réduction de la variance (page 100). Enfin, on reconsidère l'ensemble des facteurs pour une ultime étape de réduction de la fonction erreur, avec un pas divisé par 2. A la fin de chacune de ces étapes, la valeur optimale trouvée est donnée, et une netlist résultat est générée.

#### 4.4.3 RESULTATS

##### a) Interprétation des chiffres

Les résultats se présentent sous la forme de chiffres et de courbes pour faciliter leur interprétation. En considérant toujours le même exemple, les résultats sont :

```
GP: Initial Yield : e1=0.0055 s1=1.4744 yield for dots=99.47% yield for simus=46.67%
GP: Yield after SR : e2=0.0017 s2=1.4378 yield for dots=99.83% yield for simus=86.67%
GP: Yield after MT1: e3=0.0007 s3=1.4489 yield for dots=99.93% yield for simus=93.33%
GP: Yield after MT2: e4=0.0007 s4=1.4475 yield for dots=99.93% yield for simus=93.33%
```

"e" représente , l'évaluation de la fonction erreur, et "s" représente , l'écart type mesuré sur la performance. Le rendement est calculé avant toute modification de la netlist ainsi qu'après chacune des 2 (ou éventuellement 3) phases du processus d'optimisation : réduction de la variance (SR) et ajustement de la valeur moyenne de la réponse (MT). Comme cela est prévisible, c'est lors de la réduction de variance que e plus, bien que cela reste faible, mais il n'est peut être pas possible de faire mieux. Il faudrait comparer cette amélioration avec celle fournie par d'autres logiciels dans les mêmes conditions pour vérifier que l'on ne peut pas obtenir un meilleur résultat. Dans ce cas-là, il faudrait annuler l'effet de  $Ps_3$  qui nous pénalise.

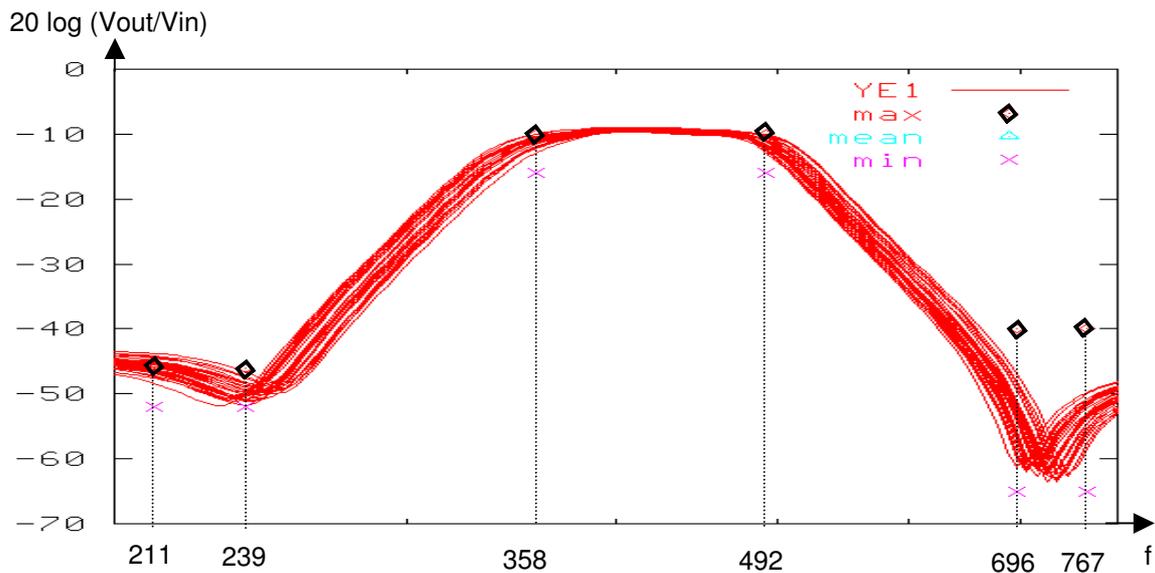
On peut suivre l'évolution du rendement en parallèle. On se rend compte que celui-ci ("yield for simus") augmente fortement, mais il faut relativiser l'amélioration apportée. "yield for simus" est calculé en considérant chaque simulation comportant au moins un point hors spécification comme hors-spécification, faisant ainsi chuter le rendement. "yield for dots" est calculé comme le rapport des points respectant les spécifications sur le nombre de points total. Le fait que celui-ci soit élevé, alors que l'analyse de Monte-Carlo ne comporte que 30 simulations de 100 points, semble suggérer que peu de spécifications sont violées, et que l'optimisation devrait amener une amélioration conséquente.

Si l'on considère maintenant l'évolution de la fonction erreur au cours du processus d'optimisation, sa plus forte régression à lieu lors de la phase de réduction de la variance. Encore une fois, ce phénomène est du à  $Ps_3$  qui est fixé de telle sorte qu'un facteur qui réduit égradant trop é. être fortement réduit lors de cette phase, et éduit encore un peu en utilisant notamment des

facteurs qui dégradent le rendement. Il est à vérifier que la dernière étape n'a pas amélioré sensiblement le rendement. Si l'on n'est pas encore satisfait de l'amélioration du rendement obtenue, on peut reprendre la netlist fournie à cette étape, et tenter de diminuer encore le pas d'itération.

### b) Interprétation des courbes

Le logiciel peut fournir un graphique [Figure 4-13] permettant de visualiser les spécifications ainsi que les courbes obtenues lors de la première analyse de Monte-Carlo réalisée pour estimer le rendement initial (30 simulations ici).

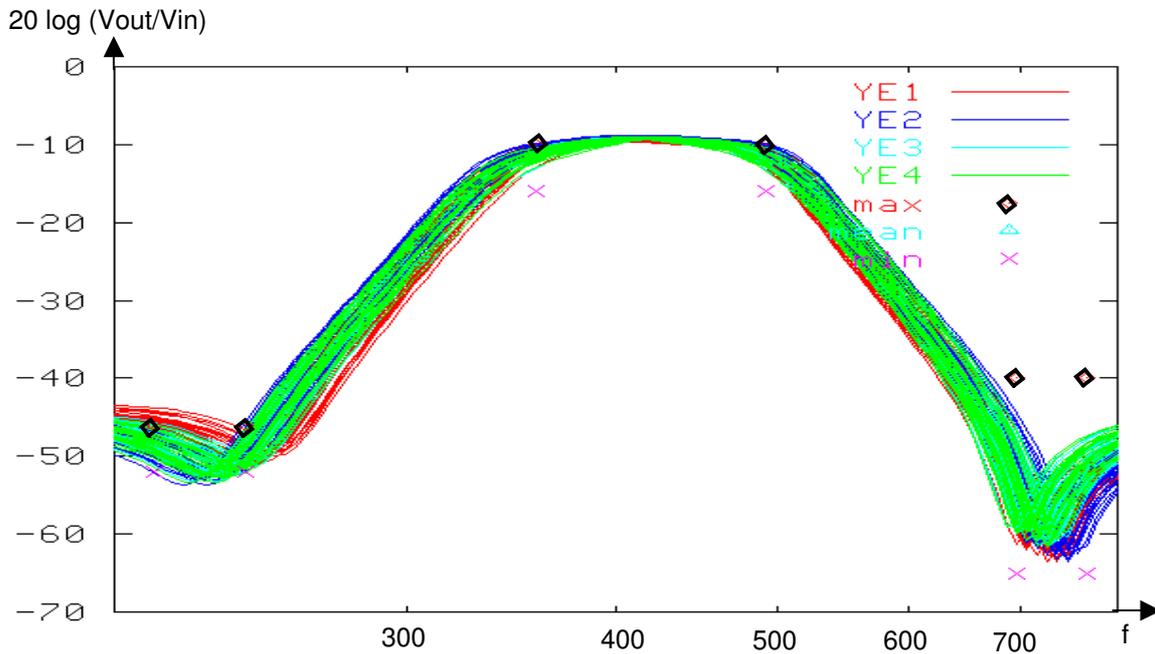


**Figure 4-13 : Tracé des spécifications et de l'analyse de Monte-Carlo pour l'estimation du rendement initial**

Sur ce graphique, les spécifications sont matérialisées par des croix, des triangles et des carrés. Les limites inférieures sont marquées par des croix, les limites supérieures par des carrés, et les points ciblés par des triangles (il n'y en a pas dans cet exemple). Comme prévu (voir page 124), on retrouve des gabarits en 6 fréquences différentes. Cette analyse de Monte-Carlo permet de constater que le gain dans la bande passante est légèrement trop élevé, et que la chute de rendement est essentiellement due au dépassement de la limite supérieure établie pour la première fréquence (210Hz).

Un autre graphe important fourni par le logiciel permet de visualiser les courbes obtenues après les analyses de Monte-Carlo ayant servi à estimer le rendement après chaque grande phase de l'algorithme d'optimisation statistique. Les courbes rouges ont servi à l'estimation du rendement initial, les bleues à l'estimation du rendement après la phase de réduction de la variance, les bleues pâles à l'estimation du rendement après la phase de

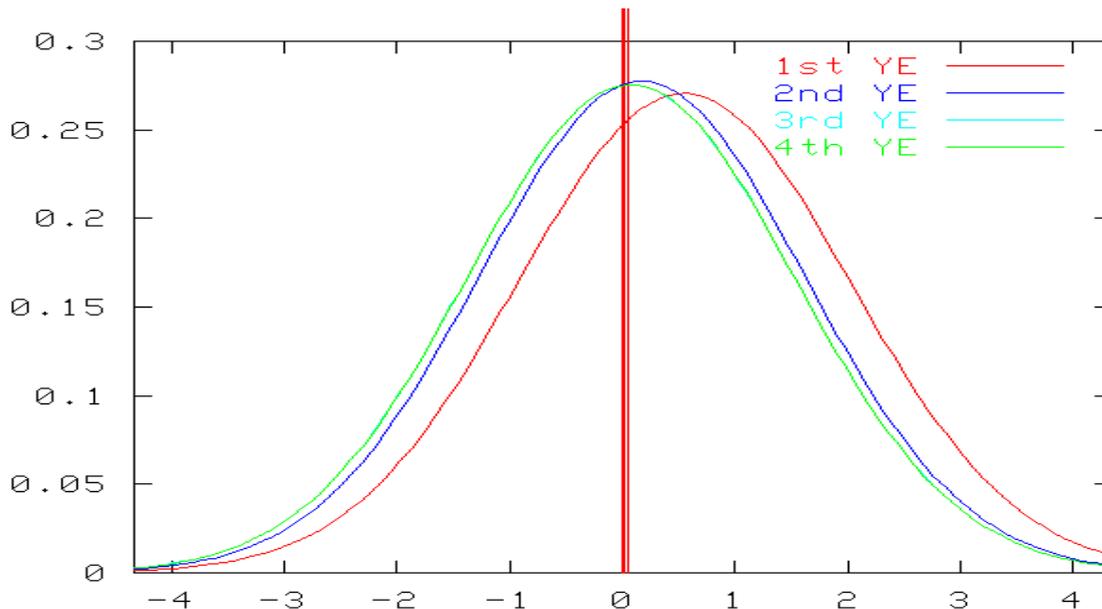
réduction de  $\sigma$  à l'estimation du rendement après la phase de réduction de  $\sigma$  à pas réduit.



**Figure 4-14 : Courbes des analyses de Monte-Carlo lors du processus d'optimisation**

Dans cet exemple, nous avons empêché la réduction  $\sigma$  telle que même une très faible dégradation de  $\sigma$  était pas tolérée. Si nous avons voulu mettre l'accent sur la réduction de la variance, nous aurions pu être plus souple à ce niveau là. Le faisceau de courbes bleues foncées (estimation du rendement après réduction de la variance) se serait alors retrouvé très éloigné de la réponse initiale. Ici, ce n'est pas le cas, et d'une estimation à l'autre, les résultats varient peu. On remarque toutefois que visiblement, la limite supérieure imposée à 210Hz est maintenant respectée.

Un troisième et dernier graphe fourni par le logiciel est le tracé de la fonction  $f(x) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{1}{2} \left( \frac{x-100*\epsilon}{\sigma} \right)^2}$  [Figure 4-15]. La position sur l'axe des abscisses est donc proportionnelle à  $\sigma$  écart type de la gaussienne est rapidement les améliorations apportées par chaque étape de l'algorithme. On constate vite que de la courbe rouge à la bleue puis la verte, la fonction erreur n'a fait que décroître. Par conséquent  $\sigma$  après avoir fortement diminué (le sommet de la courbe bleue est plus élevé que celui de la rouge), augmente légèrement. On le voit à l'aplatissement de la courbe verte par rapport à la bleue. Cela arrive car certains facteurs qui contribuent à la réduction de  $\sigma$  ont au détriment de



**Figure 4-15 : Visualisation de l'évolution de chaque paire (**

### **c) Conclusion**

L'algorithme fonctionne correctement, l'automatisation des procédures de saisie des paramètres, de lancement de simulations d'après des plans produits adaptés et de traitement des résultats permet de réaliser de nombreux tests variés sur différents circuits très rapidement. On peut ensuite comparer ces résultats avec ceux fournis par d'autres logiciels comme Circuit Surfer (Cadence) ou Aspire (Mentor Graphics). Nous nous sommes familiarisés avec ces outils et avons commencé les comparaisons, mais nous n'avons pas eu le temps de mener à terme cette tâche.

Concernant le filtre RLC passe bande, le rendement paramétrique a été nettement amélioré puisqu'il est passé de 47% à 93%. Huit paramètres de dessin et les huit paramètres de bruit faisant de ces derniers des paramètres mixtes étaient considérés lors de cette optimisation. Leur dispersion était fixée à 2% de leur valeur initiale dans la netlist. Le nombre de simulations est important : 640 pour la première phase et 288 pour la seconde. A cela, il faut rajouter les simulations dédiées à l'estimation du rendement (au moins 30), et celles arrivant éventuellement à la fin du processus, si l'on décide de diminuer le pas. Mais cela reste un nombre normal dans le domaine de l'optimisation statistique. Ajuster les paramètres de l'algorithme et améliorer certains points que l'examen des résultats mettront en lumière permettrait de gagner rapidement quelques simulations, ou au choix d'atteindre un meilleur rendement si cela est possible. Une autre amélioration envisageable est l'utilisation de toutes ces simulations (estimation du rendement et plans produits) pour avoir recours à la méthode des surfaces de réponse et tenter sans simulation supplémentaire d'améliorer encore le

rendement en travaillant sur un polynôme. On peut aussi, pour diminuer la variance de l'estimateur de Monte-Carlo, considérer certaines simulations déjà réalisées lors d'une étape précédente de l'algorithme en s'inspirant de la méthode des points communs (p. 65). Il reste donc beaucoup de travail à réaliser, mais les résultats semblent prometteurs.

#### 4.5 EXEMPLE : MICRORESONATEUR MEMS

Les peignes électrostatiques sont des structures fréquemment utilisées dans le domaine des MEMS et leur mise au point est cruciale [LEE98] [YE98] [JOHN95]. Ils peuvent servir d'actionneurs ou de capteurs (accéléromètres...) et sont à la base de systèmes de type résonateur, gyroscope [RENK99] ou filtre [NGUY98] [JING00] [LIN98]. De nombreux brevets ont été déposés dans ce dernier cas [DELP-net]. Dans cet exemple, nous nous attacherons à contrôler précisément le déplacement d'une telle structure lorsqu'elle est soumise à un signal de commande donné.

Pour valider simplement l'approche que nous avons choisie pour l'optimisation statistique du rendement, une analyse de Monte-Carlo peut suffire. Si cette dernière confirme que la netlist dont les paramètres ont été optimisés affiche un rendement plus important que l'original, c'est que nous sommes sur la bonne voie. Cela, sauf cas spéciaux sur lesquels nous reviendrons, a été vérifié. En parallèle, il faut aussi vérifier que le rendement *réel* est aussi amélioré. L'obtention de circuits intégrés dans le cadre d'une fabrication multi-projets étant assez longue, nous nous sommes attachés à réaliser une telle structure assez tôt, pour avoir le temps de caractériser statistiquement la technologie et choisir l'une des différentes formes de MEMS testées. Différentes structures ont été envisagées dans le but de réaliser à long terme un résonateur à haute fréquence. La [AMS-net]). Il s'agit d'une technologie pour circuits intégrés CMOS, utilisable pour réaliser des MEMS [MUKH98] : en fin de fabrication, une gravure est réalisée pour libérer les parties mobiles. Cette étape [Figure 4-16] est réalisée à l'université américaine Carnegie Mellon University [CMU-net].

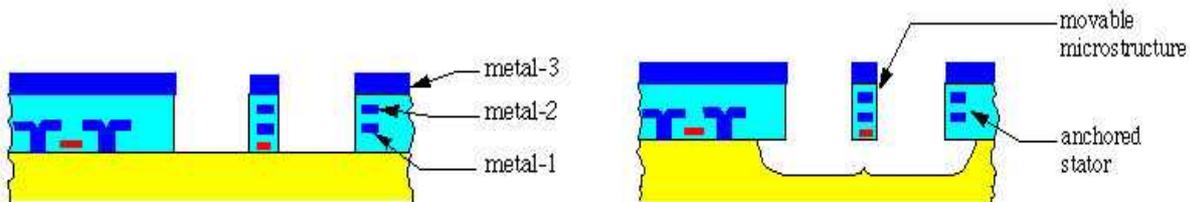
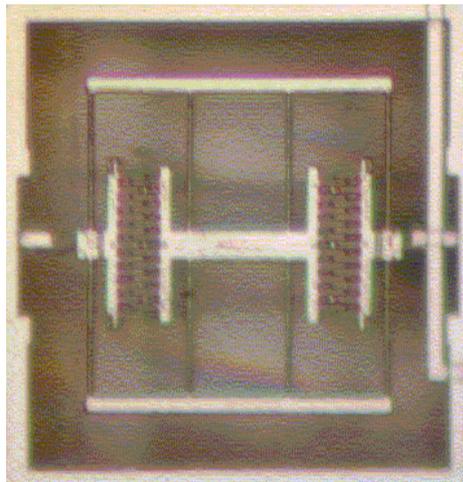


Figure 4-16 : Nécessité d'une étape *post-process* pour libérer les structures

Des tests ont été réalisés au CIME (Centre Inter-universitaire de Micro-Electronique) de Grenoble. Nous avons pu nous y assurer que les structures étaient bien mobiles et que le câblage était bon. Mais nous n'avons pas disposé sur place du matériel de mesure nous permettant de cerner l'aspect statistique de la technologie. La Figure 4-17 montre une structure à peignes électrostatiques conçue pour se comporter comme un filtre passe-bande dont la fréquence de résonance est imposée par sa géométrie et les caractéristiques des matériaux qui la composent. Plus de détails au sujet de cette étude sont donnés dans l'annexe III.



**Figure 4-17 : Filtre MEMS réalisé avec une technologie AMS, testé au CIME**

La phase de recherche d'une structure performante est longue, et il était prévu dans un premier temps de trouver une structure répondant à nos attentes (fréquence de résonance, tension d'actuation...), puis de la décliner en plusieurs variantes, et enfin seulement de modéliser précisément la plus prometteuse pour l'optimiser grâce au logiciel. La puce réalisée comporte une quinzaine de structures différentes, avec deux formes principales de filtres pour lesquels différentes dimensions sont testées. Cette même structure sert de base à une équipe de Duke University (U.S.A.) dans le cadre de l'optimisation du rendement des MEMS [DEWE98] [REN01]. Finalement, n'ayant pu faire fabriquer qu'une seule puce, cette démarche n'a pu aboutir. C'est pourquoi cette structure n'est pas tout à fait la même que celle que nous avons optimisée dans notre exemple et qui a l'avantage d'être rapidement modélisée.

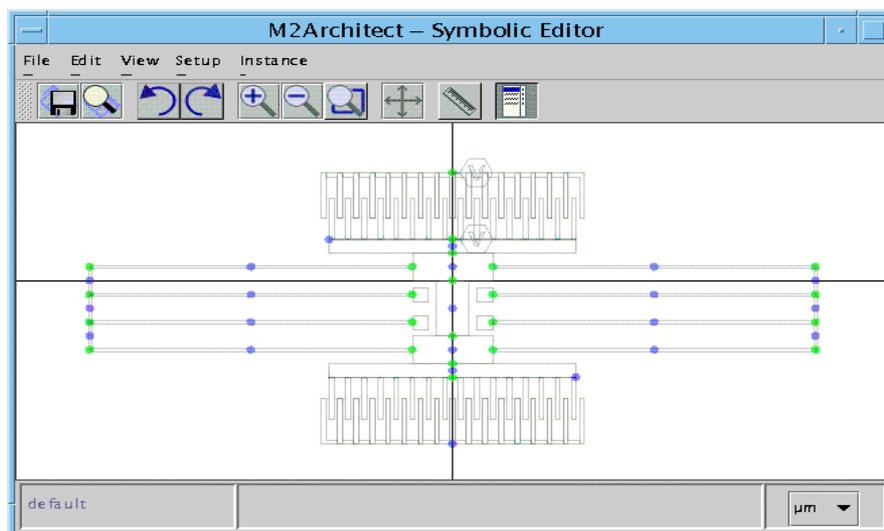
#### 4.5.1 SAISIE DU PROBLEME

Dans l'exemple précédent (filtre RLC), nous réalisons une optimisation statistique sur la base d'une performance évaluée lors d'une analyse fréquentielle. Un des objectifs vers lequel nous voulons tendre étant l'optimisation multi-objectifs, nous avons envisagé la

possibilité de traiter lors d'une même session des analyses transitoires. L'exemple suivant fera donc l'objet d'une optimisation statistique où les spécifications sont précisées dans le domaine temporel.

### a) Topologie du MEMS

Une netlist décrivant la structure qui nous intéresse est générée avec l'outil MemMaster [MOUL02]. Chaque composant de base, par exemple une poutre, est associé à un modèle comportemental Verilog-A. Ses paramètres pourront donc être modifiés au cours de l'optimisation par le logiciel.



**Figure 4-18 : Elaboration d'une structure MEMS avec MemMaster**

La netlist est créée par le logiciel d'après la vue symbolique de la structure. On peut lui associer un fichier décrivant la technologie utilisée, ce qui permettra par la suite de générer automatiquement les masques de cette structure.

La netlist est extraite du schéma précédent et se présente sous la forme suivante :

```

simulator lang = spectre
ahdli1 (net1 net2 net3 net4 net5 net6 0 0 0) PLATE_NS_UxUyMz width=4.0E-5 length=2.0E-5
ahdli2 (net7 net8 net9 net1 net2 net3 net10 net11 net12 net13 net14 net15 0 0 0) PLATE_NSEO_UxUyMz
width=2.0E-5 length=5.0E-5
ahdli3 (net4 net5 net6 net16 net17 net18 net19 net20 net21 net22 net23 net24 0 0 0) PLATE_NSEO_UxUyMz
width=2.0E-5 length=5.0E-5
ahdli4 (net25 net26 net27 net7 net8 net9 0 0 0) PLATE_NS_UxUyMz width=1.0E-5 length=1.53E-4
ahdli5 (net16 net17 net18 net28 net29 net30 0 0 0) PLATE_NS_UxUyMz width=1.0E-5 length=1.53E-4
ahdli6 (net31 net32 net33 net13 net14 net15 0 0 0) BEAM_UxUyMz width=2.0E-6 length=2.0E-4 angle=0.0
ahdli7 (net31 net32 net33 net34 net35 net36 0 0 0) BEAM_UxUyMz width=5.0E-6 length=2.0E-5 angle=-90.0
ahdli8 (net34 net35 net36 net37 net38 net39 0 0 0) BEAM_UxUyMz width=5.0E-6 length=2.0E-5 angle=-90.0
ahdli9 (net37 net38 net39 net40 net41 net42 0 0 0) BEAM_UxUyMz width=5.0E-6 length=2.0E-5 angle=-90.0
    
```

```
ahdli10 (net34 net35 net36 net43 net44 net45 0 0 0) BEAM_UxUyMz width=2.0E-6 length=2.0E-4 angle=0.0
ahdli11 (net37 net38 net39 net46 net47 net48 0 0 0) BEAM_UxUyMz width=2.0E-6 length=2.0E-4 angle=0.0
ahdli12 (net40 net41 net42 net22 net23 net24 0 0 0) BEAM_UxUyMz width=2.0E-6 length=2.0E-4 angle=0.0
ahdli13 (net10 net11 net12 net49 net50 net51 0 0 0) BEAM_UxUyMz width=2.0E-6 length=2.0E-4 angle=0.0
ahdli14 (net49 net50 net51 net52 net53 net54 0 0 0) BEAM_UxUyMz width=5.0E-6 length=2.0E-5 angle=-90.0
ahdli15 (net52 net53 net54 net55 net56 net57 0 0 0) BEAM_UxUyMz width=5.0E-6 length=2.0E-5 angle=-90.0
ahdli16 (net55 net56 net57 net58 net59 net60 0 0 0) BEAM_UxUyMz width=5.0E-6 length=2.0E-5 angle=-90.0
ahdli17 (net61 net62 net63 net52 net53 net54 0 0 0) BEAM_UxUyMz width=2.0E-6 length=2.0E-4 angle=0.0
ahdli18 (net64 net65 net66 net55 net56 net57 0 0 0) BEAM_UxUyMz width=2.0E-6 length=2.0E-4 angle=0.0
ahdli19 (net19 net20 net21 net58 net59 net60 0 0 0) BEAM_UxUyMz width=2.0E-6 length=2.0E-4 angle=0.0
ahdli20 (net25 net26 net27 net67 0 0 0 net68) COMB_UxUyMzV nbfingers=16.0 fgap=2.0E-6 overlap=1.5E-5
fwidth=3.0E-6 flength=3.0E-5 angle=0.0
ahdli21 (net28 net29 net30 0 0 0 0 0) COMB_UxUyMzV nbfingers=16.0 overlap=1.5E-5 fgap=2.0E-6 fwidth=3.0E-
6 flength=3.0E-5 angle=180.0
ahdli22 (net61 net62 net63) ANCHOR_UxUyMz width=1.0E-5 length=1.0E-5 angle=0.0
ahdli23 (net64 net65 net66) ANCHOR_UxUyMz width=1.0E-5 length=1.0E-5 angle=0.0
ahdli24 (net43 net44 net45) ANCHOR_UxUyMz width=1.0E-5 length=1.0E-5 angle=180.0
ahdli25 (net46 net47 net48) ANCHOR_UxUyMz width=1.0E-5 length=1.0E-5 angle=180.0
ahdli26 (net67) V_CST ac_mag=0.0 dc=0.0 ac_phase=0.0 value=0.0
ahdli27 (net68) V_SQR tfall=1.0E-9 freq=2500.0 ac_phase=0.0 offset=2.5 trise=1.0E-9 ratio=0.8 ampl=10.0
ac_mag=5.0 dc=0.0
```

On y discerne les modèles utilisés des poutres (beam), des plaques (plate), des ancrés (anchor) et des peignes électrostatiques (comb). Il faut maintenant définir les paramètres impliqués dans l'optimisation, les paramètres propres à l'algorithme conservent leurs valeurs, fixées lors de l'exemple précédent (page 124).

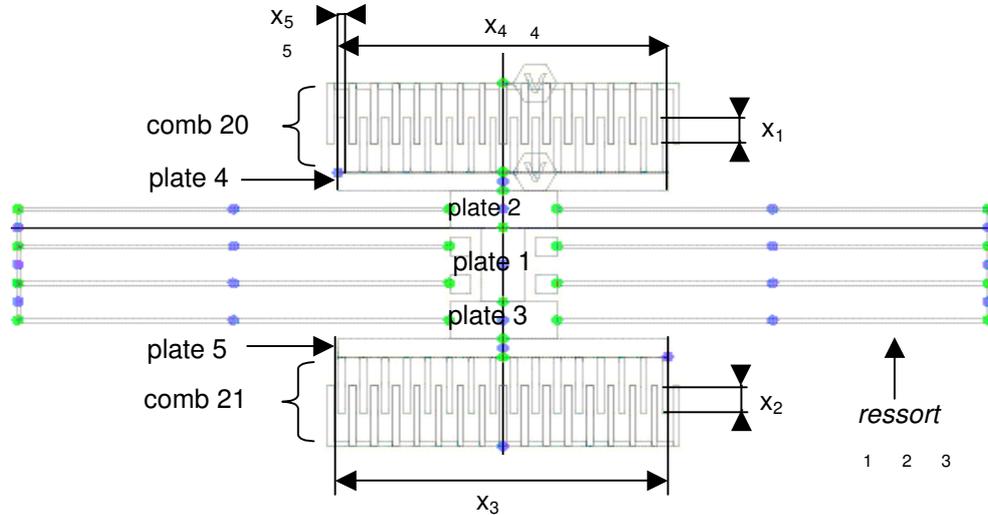
### **b) Choix des paramètres et de leurs caractéristiques**

Il serait intéressant d'utiliser ici les paramètres de bruit intégré, mais nous n'avons pas eu le temps de modifier et de tester l'ensemble des modèles pour satisfaire à cette particularité décrite au § 4.1.3. Comme les analyses transitoires sont beaucoup plus longues à réaliser que les analyses fréquentielles, nous nous contenterons de 5 paramètres de bruit et de 5 paramètres de dessin, ce qui nous permettra d'utiliser des plans d'expériences de résolution V avec la table orthogonale de Taguchi L16 (voir Tableau 3-2 page 107). Les paramètres de dessin (notés sur la Figure 4-19) des peignes sont :

- l'*overlap* ( $x_1$  et  $x_2$ ), distance de recouvrement de deux doigts en regard et de potentiels différents
- la longueur des plaques 4 et 5 (*plate*) qui soutiennent les doigts des peignes ( $x_3$  et  $x_4$ )
- l'épaisseur des doigts du peigne 20 (*comb*) ( $x_5$ )

Les paramètres de bruit sont :

- la largeur de trois des poutres (*beam*) composant les "ressorts" de la structure ( $x_1, x_2, x_3$ )
- la longueur de la plaque 4 ( $x_4$ )
- la largeur des doigts du peigne 20 ( $x_5$ )



**Figure 4-19 : Paramètres de dessin et de bruit du MEMS**

Les paramètres reportés sur la figure précédente se retrouvent dans la netlist, et dans les fichiers de description des paramètres sous la forme suivante :

|         |                   |         |        |         |
|---------|-------------------|---------|--------|---------|
| $x_1$ : | D_ahdli20_overlap | 10e-6   | 20e-6  | 5e-6    |
| $x_2$ : | D_ahdli21_overlap | 10e-6   | 20e-6  | 5e-6    |
| $x_3$ : | D_ahdli5_length   | 1.5e-4  | 1.9e-4 | 0.01e-4 |
| $x_4$ : | D_ahdli4_length   | 1.5e-4  | 1.9e-4 | 0.01e-4 |
| $x_5$ : | D_ahdli20_fwidth  | 2e-6    | 4e-6   | 0.1e-6  |
| $x_1$ : | N_ahdli13_width   | 0.02e-6 | 0      | 2       |
| $x_2$ : | N_ahdli14_width   | 0.02e-6 | 0      | 2       |
| $x_3$ : | N_ahdli19_width   | 0.02e-6 | 0      | 2       |
| $x_4$ : | N_ahdli4_length   | 3e-6    | 0      | 2       |
| $x_5$ : | N_ahdli20_fwidth  | 0.05e-6 | 0      | 3       |

### c) Choix des spécifications

Les variations des performances sont beaucoup moins marquées en analyse temporelle que fréquentielle, nous ne considérerons donc que deux points sensibles :

```
tran_net2_lat-res tran net2 1 0 1e-3 1e-5 mag
0.0      NaN    NaN    NaN
1.0E-5   NaN    NaN    NaN
2.0E-5   NaN    NaN    NaN
...
2.4E-4   NaN    NaN    NaN
2.5E-4   NaN    1.13e-6 1.18e-6
```

```

2.6E-4  NaN  NaN  NaN
...
4.9E-4  NaN  NaN  NaN
5.0E-4  NaN  -1e-8  1e-8
5.1E-4  NaN  NaN  NaN
5.2E-4  NaN  NaN  NaN
8.2E-4  NaN  NaN  NaN
0.0010  NaN  NaN  NaN

```

#### 4.5.2 SUIVI DU PROCESSUS D'OPTIMISATION

Comme pour l'exemple précédent, nous pouvons suivre le cheminement de la valeur numérique de chaque paramètre au cours du processus d'optimisation, par exemple  $x_2$  (ahdli21\_overlap values) :

```

initial value=1.5E-5  MIN=1.0E-5  MAX=2.0E-5  STEP=5.0E-7
All values have been updated by FactorCretor.setFactorsAttributes: new values are MIN=1.2E-5
MAX=1.8E-5  STEP=1.5000000000000002E-7

```

##### ***Sigma reduction***

###### **LoopA=1**

```

L1=1.4925E-5  L2=1.5075E-5
E1=0.0025188525692361667  E2=0.0025188525643162433  deltaE=-4.919923409646421E-12
S1=6.215658299772485E-9  S2=6.215583588627465E-9  deltaS=-7.471114502028875E-14

```

###### **LoopA=2**

```

L1=1.5225E-5  L2=1.5375E-5
E1=0.002517986095809811  E2=0.0025179944875556187  deltaE=8.391745807793949E-9
S1=6.2171110159476947E-9  S2=6.2182551999493535E-9  deltaS=1.1450404724065412E-12

```

###### **LoopA=3**

```

L1=1.5125E-5  L2=1.5175000000000001E-5
E1=0.002517409268388163  E2=0.002517409268388163  deltaE=0.0
S1=6.213450126913867E-9  S2=6.213450126913867E-9  deltaS=0.0

```

DF is fixed at sigmaMin: stopstep was=true

###### **NoptA=1.5125E-5**

##### ***Mean Tuning***

###### **LoopB=1**

```

L1=1.505E-5  L2=1.52E-5
E1=0.0025167363923151703  E2=0.002516721265246034  deltaE=-1.5127069136373905E-8
S1=6.2230614369706955E-9  S2=6.222760662242437E-9  deltaS=-3.0077472825827296E-13

```

DF is fixed at epsMin: deltaEps was not important enough:Pe1%.Max(eps2,eps1)=2.51673639231E-8

###### **NoptB=1.52E-5**

Mean Tuning : all DF are considered a last time with loopBMax=3

###### **LoopB=1**

```

L1=1.5162500000000001E-5  L2=1.52375E-5
E1=0.0025087242599277615  E2=0.0025087255008823164  deltaE=1.2409545548271173E-9
S1=6.205865775654216E-9  S2=6.206445322673645E-9  deltaS=5.795470194294804E-13

```

DF is fixed at epsMin: deltaEps was not important enough:Pe1%.Max(eps2,eps1)=2.50872550088E-8

###### **NoptB=1.5162500000000001E-5**

Nous en concluons qu'il est conseillé de légèrement augmenter l'overlap (de 0, pour améliorer la robustesse de la structure.



#### 4.6 EXEMPLE : INDUCTANCE MEMS

On attend beaucoup des inductances MEMS pour les applications RF, notamment en téléphonie mobile, car elles peuvent atteindre des facteurs de qualité très élevés et être intégrées au reste de l'électronique, limitant les parasites dus à la connectique et le volume utilisé [GAMM99].

Le concepteur qui désire utiliser une bobine MEMS spécifie la valeur maximale souhaitée pour son facteur de qualité ( $Q$ ) à une fréquence et une inductance ( $L$  en nH) données, avec une très grosse contrainte : la surface tolérée [Figure 4-21]. Le facteur est d'autant plus élevé que les pertes résistives et capacitives sont faibles.

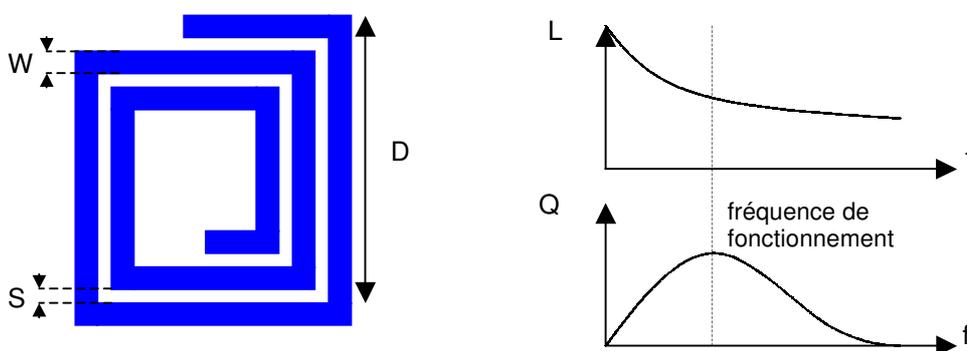


Figure 4-21 : Paramètres géométriques d'une inductance MEMS

$S$  représente la distance inter-spires,  $W$  la largeur de la piste,  $D$  le diamètre de la bobine. La réponse de cette structure dépend fortement de phénomènes physiques complexes étroitement liés aux paramètres technologiques [Figure 4-22], ce qui la rend intéressante comme structure de test pour pratiquer une optimisation statistique.

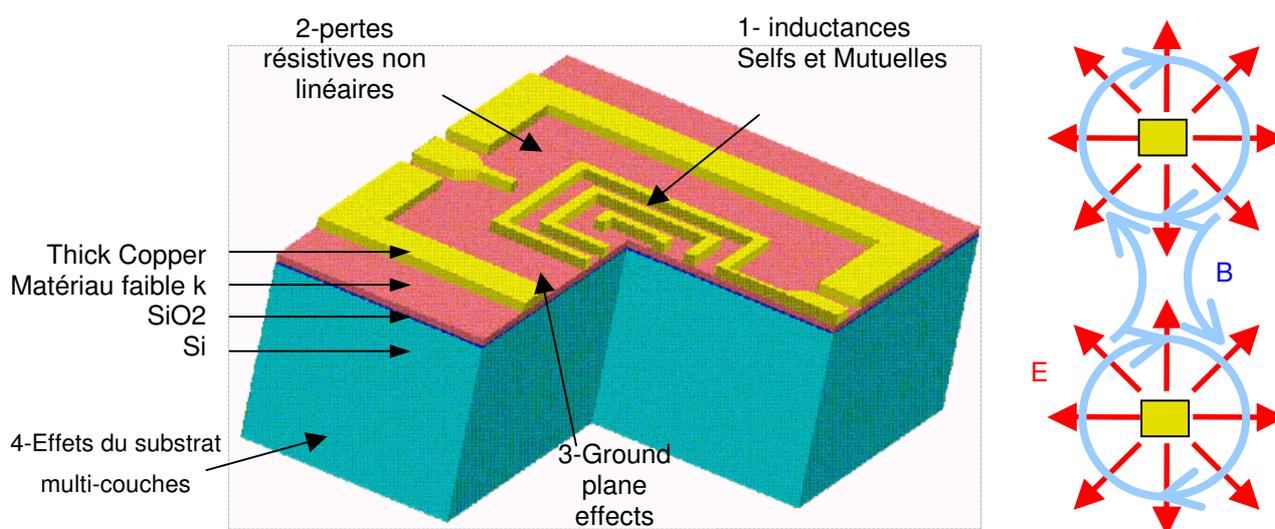


Figure 4-22 : Principaux phénomènes physiques liés à une inductance

Un courant variable circulant dans le conducteur créé un champ électromagnétique. L'inductance propre (*self-inductance*) représente l'énergie stockée dans le champ magnétique lorsque ce dernier est créé par un courant alternatif. L'inductance mutuelle apparaît lorsque les champs magnétiques générés par le courant circulant dans deux conducteurs donnent naissance à un champ global qui peut les perturber. Le concepteur peut modifier la valeur de l'inductance propre en travaillant sur les paramètres géométriques de la structure tels que hauteur, largeur (W) et longueur du conducteur. Il peut modifier l'inductance mutuelle en traitant la distance inter-spires (S).

La densité volumique de courant décroît lorsque l'on s'éloigne de la surface du conducteur (effet de peau). Pour une densité donnée, l'épaisseur correspondante est fonction de la fréquence du courant alternatif et du matériau utilisé comme conducteur. La section effective du conducteur diminue ainsi avec la fréquence de fonctionnement, augmentant les pertes résistives. Pour les limiter, le concepteur peut donc : diminuer la fréquence de fonctionnement et la longueur du conducteur, augmenter la section du conducteur (largueur W) ou la conductivité du matériau utilisé.

Les pertes dans le substrat ont deux origines : sa conductivité et sa permittivité. Lorsque le substrat est conducteur (cas du silicium), une partie du champ magnétique est "absorbé" par les courants induits en son sein, une partie de l'énergie magnétique est ainsi dissipée dans le substrat et ce phénomène augmente avec la fréquence du courant.

Il existe aussi une capacité parasite, entre le conducteur et la masse électrique. Elle est la cause de pertes supplémentaires en hautes fréquences, et peut être réduite en limitant la surface de contact entre le conducteur et le substrat, ainsi qu' en réduisant la permittivité du substrat ou en augmentant son épaisseur ( $C = \text{surface} \cdot \text{permittivité} / \text{épaisseur}$ ). On peut aussi insérer un matériau à faible permittivité entre le conducteur et le substrat pour limiter les pertes dues à cette capacité [Figure 4-23].

Le but, pour le technologue puis le concepteur, est d'obtenir un facteur de qualité maximum pour une inductance L. On peut résumer les contraintes de conception ainsi :

Pour augmenter la valeur de L :

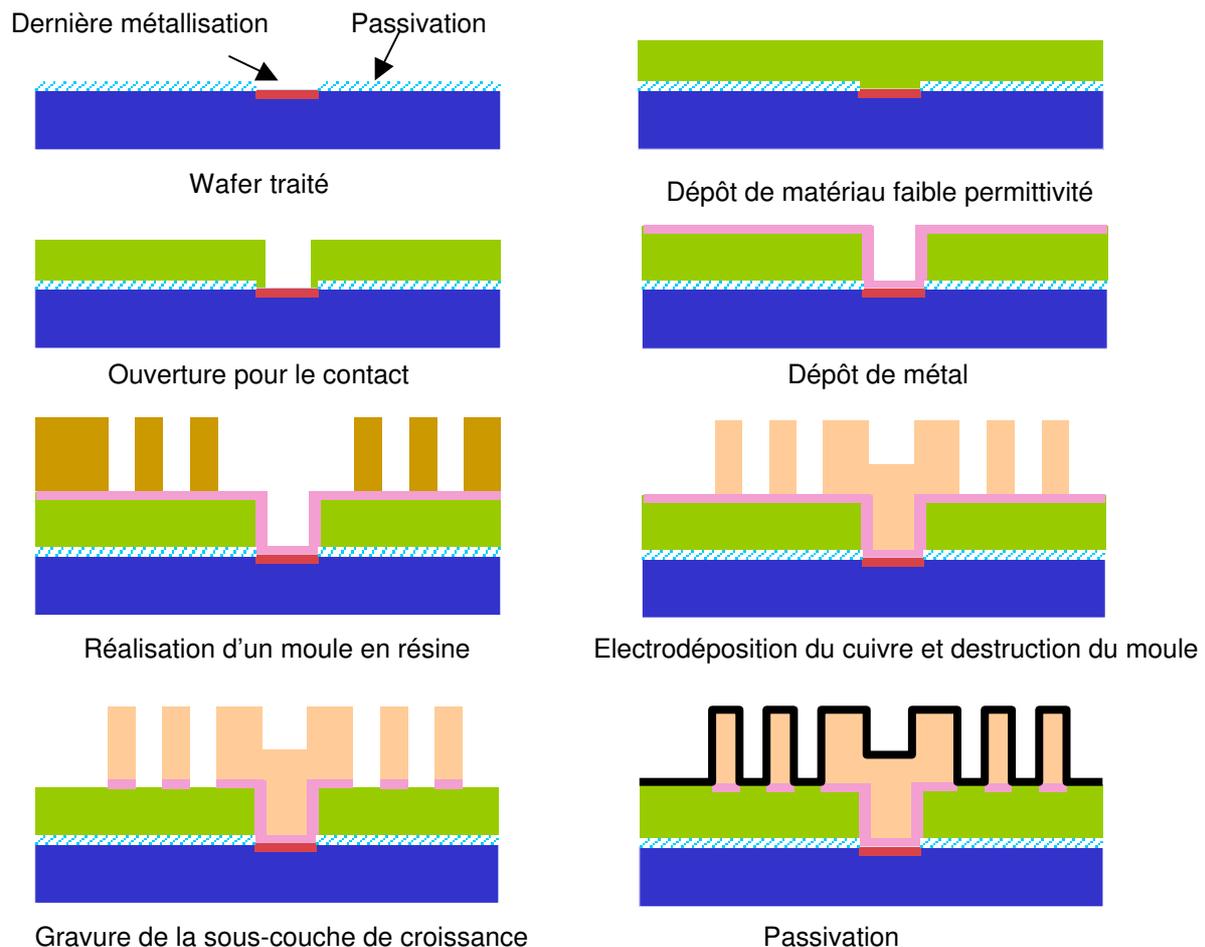
- augmenter la longueur du conducteur
- réduire la section du conducteur
- réduire la distance inter-spires (W)

Pour limiter les pertes résistives :

- augmenter la section du conducteur
- travailler à faible fréquence

Pour limiter les pertes dans le substrat :

- réduire la surface du conducteur
- utiliser un substrat de faible permittivité



**Figure 4-23 : Description d'une technologie MEMS pour une inductance**

Vue la difficulté à modéliser convenablement une telle structure de manière comportementale afin de la simuler et de l'optimiser, l'approche habituelle est de fabriquer de nombreuses variantes, de sélectionner les plus performantes et de construire une librairie autour de ces structures. Mais cela ne permet l'utilisation que de certaines valeurs d'inductance, et laisse de coté les circuits de type full-custom.

Une seconde approche est de modéliser analytiquement les inductances. Leur réponse est évaluée par des outils de simulation électromagnétique, et l'ensemble des résultats

permet de construire un polynôme dont les coefficients sont les paramètres de dessin des structures. Ce polynôme modélise la réponse du MEMS. On peut par exemple utiliser les plans d'expériences avec la méthode des surfaces de réponse. Par la suite, la fabrication et la caractérisation de quelques structures permettent d'affiner les coefficients du polynôme. Ce polynôme sera le modèle analytique décrivant l'inductance que nous placerons comme une boîte noire dans un circuit à optimiser. Les paramètres de dessin qui y figurent seront les facteurs de dessin que nous pourrions choisir d'optimiser.

L'idéal serait de réaliser une double optimisation : à la fois sur Q et sur L, à des fréquences de l'ordre de 2GHz. Trois obstacles nous en empêchent actuellement. Tout d'abord, le fait que 2 performances soient à optimiser simultanément. Notre stratégie pour l'optimisation multi-objectifs a été énoncée, mais son implémentation est toujours en cours. Ensuite, un modèle qui permet l'estimation de Q(f). Ce modèle existe, mais la dépendance du modèle envers la fréquence de fonctionnement n'est pas aisée à mettre en place dans un modèle Verilog-A. Pour finir, nous avons écrit un *parser* permettant à notre logiciel d'optimisation de traiter des netlists écrites en Verilog-A pour le simulateur Spectre, et ce dernier n'est pas prévu pour des simulations à de telles fréquences de fonctionnement. Nous ne sommes donc pas dans les conditions idéales pour optimiser les performances les plus intéressantes d'une telle structure. Nous pouvons tout de même démontrer la bonne tenue de notre approche pour d'autres performances. Si les résultats s'avèrent intéressants, on pourra alors envisager de consacrer plus d'efforts à adapter le logiciel à un plus grand nombre de simulateurs et de langages de description de circuits.

#### **4.7 VARIANTE : OPTIMISEUR NOMINAL**

Si l'on ne considère que la seconde étape de l'algorithme (p.100), on s'attachera à réduire l'erreur entre la courbe cible et celle obtenue après simulation. Dans le cadre d'une optimisation nominale, on ne considère pas l'aspect statistique des paramètres de bruit. Ces derniers sont donc fixés à leur valeur moyenne. On n'utilisera donc plus les plans produits de Taguchi, puisque le plan bruit n'a plus de raison d'être. Par contre, le plan principal sera utilisé comme un plan d'expériences classique. Nous prendrons alors soin de fixer  $pas2=2*pas1$ , tels qu'ils sont définis p.96. Cette possibilité de l'algorithme n'a pas encore été explorée, mais peut être rapidement évaluée puisqu'il ne s'agit que d'un sous-ensemble de l'implémentation déjà réalisée. Elle permettrait de mener à bien la phase d'optimisation nominale qui augmente l'efficacité de l'optimisation statistique à suivre.

Dans ce chapitre, nous sommes revenus sur la modélisation même des MEMS et sur le choix de nos modèles. Les contraintes de l'informatique ont été abordées et la structure du logiciel détaillée. Enfin des exemples ont été traités pour démontrer la bonne tenue de l'algorithme et le bon comportement du logiciel. Deux types d'analyses ont pu être ainsi abordés : l'analyse fréquentielle et l'analyse temporelle. Nous évoquons aussi la possibilité de décliner facilement un optimiseur nominal basé sur la structure déjà implémentée.

## REFERENCES

- [AMS-net] Austria Micro Systems  
[http://www.austriamicrosystems.com/03products/core\\_ip\\_sensors.htm](http://www.austriamicrosystems.com/03products/core_ip_sensors.htm)
- [BAIDYa99] Bikram Baidya, "MEMS extraction", M.S. Th. : Elec. : Carnegie Mellon University : 1999 : 36 p.
- [BAIDYb99] B. Baidya, S.K. Gupta and T. Mukherjee, "MEMS Component Extraction", technical Proceedings of the 1999 International Conference on Modeling and Simulation of Microsystems, Puerto Rico-U.S.A., pp. 143-146.
- [BONI94] Boning, D. S., T. Maung, J. Chung, K.-J. Chang, S.-Y. Oh, D. Bartelink, "Statistical metrology of interlevel dielectric thickness variation", Proceedings of the SPIE Symposium on Microelectronics Manufacturing, Austin, U.S.A., 1994, SPIE Vol. 2334, pp. 316-327.
- [BONI96] D.S. Boning, J.E. Chung, "Statistical metrology: understanding spatial variation in semiconductor manufacturing", Microelectronic Manufacturing Yield, Reliability, and Failure Analysis II, SPIE 1996 Symposium on Microelectronic Manufacturing, Austin, U.S.A.
- [BONI99] D. S. Boning, J. Stefani, and S. W. Butler, "Statistical Methods in Semiconductor Manufacturing," in Encyclopedia of Electrical Engineering, J.G. Webster, Ed., Wiley, 1999.
- [BONI01] D.S. Boning, S. Nassif, "Models of process variations in device and interconnect", dans "Design of High-Performance Microprocessor Circuits", Wiley-IEEE Press, 2001, chapitre 6, pp. 98-116.
- [CARL97] E.T. Carlen, C.H. Mastrangelo, "An experimental methodology for the estimation of spatially correlated parametric yield in thin film devices", 1997 SISPAD Conference, Cambridge, MA.
- [CHAN95] E. Chang et al., "Using a Statistical Metrology Framework to Identify Random and Systematic Sources of Intra-Die ILD Thickness Variation for CMP Processes", 1995 International Electron Devices Meeting, Wash. D.C., pp. 499-502
- [CMU-net] Carnegie Mellon University <http://www.ece.cmu.edu/~mems/>
- [DELP-net] Recensement des brevets sous <http://www.delphion.com/>, notamment :

- \*C.T.-C. Nguyen and R. Howe, US5955932: "Q-controlled microresonators and tunable electric filters using such resonators", 1999.
- \*L. Lin, C.T.-C.Nguyen, R.T. Howe and A.P. Pisano, US5537083 : "Microelectromechanical signal processors", 1996.
- \*L. Lin, C.T.-C. Nguyen, R.T. Howe and A.P. Pisano, US5589082 : "Microelectromechanical signal processor fabrication", 1996.
- \*W.C. Tang and R.T. Howe, US5025346 : "Laterally driven resonant microstructures", 1991.
- \*S.G. Adams, Y.C. Wang, N.C. Macdonald and J.S. Thorp, US5914553 : "Multistable tunable micromechanical resonators", 1999.
- [DEWE98] Allen Dewey, Tianhao Zhang, Hong Ren, and Junhua Wu, "Parametric regression analysis of microelectromechanical resonator", Proceeding of Behaviour Modeling and Simulations '98, Orlando, Florida.
- [DIAZ97] E.V.Saavedra Diaz, K.G. McCarthy, D.B.M. Klaassen, A. Mathewson, "Efficient Parameter Extraction and Statistical Analysis for a 0.25um low-power CMOS Process", proceedings of the 27th European Solid-State Device Research Conference (ESSDERC), 1997, pp. 656-659.
- [GABB98] Lynn Daniel Gabbay, "Computer aided macromodeling for MEMS", Th. : Elec. : Massachusetts Institute of Technology : 1998 : 87 p.
- [GAMM99] P. L. Gammel, B. P. Barber, V.Lubecke, N. Belk, M.R. Frei, "Design, test, and simulation of self-assembled micromachined rf inductors", Design, Test, and Microfabrication of MEMS and MOEMS '99, Paris, SPIE Vol. 3680, pp.582-591.
- [HANN99] J.P. Hanna, R.G. Hillman, "A common basis for mixed-technology micro-system modeling", proceedings of Modeling and Simulation of Micosystems '99, San Juan, USA, avril 1999, disponible sur <http://www.cr.org>.
- [HEAL99] S. Healy, E. Horan, K. McCarthy, A. Mathewson, Z. Ning, E. Rombouts, W. Vanderbauwhede, M. Tack, "Implementation of Statistical Characterisation and Design Techniques for an Industrial 0.5um CMOS Technology", proceedings of the IEEE 1999 Int. Conference on Microelectronic Test Structures, pp. 227-232.
- [JING00] Q. Jing, H. Luo, T. Mukherjee, L.R. Carley and G.K. Fedder, "CMOS micromechanical bandpass filter design using a hierarchical MEMS circuit library", in the 13th Annual Int'l Micro Electro Mechanical Systems Conference, Miyazaki, Japan, 2000, pp. 187-192.
- [JOHN95] W.A.Johnson and L.K.Warne, "Electrophysics of micromechanical comb actuators", Journal of Microelectromechanical Systems, Mar. 1995, v.4, no.1, pp.49-59.
- [LEE98] K.B. Lee, "A triangular electrostatic comb array for micromechanical resonant frequency tuning", SENSORS AND ACTUATORS A: PHYSICAL, Vol. : 70, 1998, pp. 112-117.
- [LIN98] L. Lin et al., "Microelectromechanical Filters for Signal Processing", Journal of Microelectromechanical Systems, vol. 7, No. 3, Sep. 1998, pp. 286-294.

- [MCCA98] K.G. McCarthy, E.V. Saavedra Diaz, D.B.M. Klaassen, A.Mathewson, "Statistical Characterisation and Analysis of 0.18um Low-Power CMOS Process", proceedings of the IEEE 1998 Int. Conference on Microelectronic Test Structures (ICMTS), 1998, pp. 127-131.
- [MEHR00] V. Mehrotra et al., "A methodology for modeling the effects of systematic within-die interconnect and device variations on circuit performances", DAC 2000, Los Angeles, California.
- [MOUL02] D. Moulinier, P. Nachtergaele, M-P. Brutails, S. Bergeon, S. Maza, Ch. Hui Bon Hoa, "Efficient MEMS design methodology integrating finite elements based macro models into a nodal approach MEMS component design tool", Proceedings of Design, Test, Integration and Packaging of MEMS/MOEMS : DTIP 2002, Cannes, 9 p.
- [MUKH98] T. Mukherjee, S. Iyer, G.K. Fedder, "Optimization-based synthesis of microresonators", in "Sensors and Actuators A", ed. Elsevier, 1998, pp. 118-127.
- [NGUY98] C.T.-C. Nguyen, "Frequency-Selective MEMS for Miniaturized Communication Devices", 1998 IEEE Aerospace Conference Proceedings, vol. 1, pp. 445-460.
- [NGUY99] C. T.-C. Nguyen, "Micromechanical components for miniaturized low-power communications (invited)," Proceedings, 1999 IEEE MTT-S International Microwave Symposium RF MEMS Workshop (on Micro-electromechanical Devices for RF Systems: Their Construction, Reliability, and Application), Anaheim, California, pp. 48-77.
- [NMRC] National Microelectronics Research Center (Irlande) <http://www.nmrc.ie/>
- [OUMA96] Ouma, D., B. Stine, R. Divecha, D. Boning, J. Chung, I. Ali, and M. Islamraja,, "Using Variation Decomposition Analysis to Determine the Effect of Process on Wafer and Die-Level Uniformities in CMP", First International Symposium on Chemical Mechanical Planarization (CMP) in IC Device Manufacturing, Vol. 96-22, pp. 164-175, 190<sup>th</sup> Electrochemical Society Meeting, San Antonio, TX, 1996.
- [REN01] H. Ren, A. Jgo, R.B. Fair, "Statistical optimal design of microelectromechanical system (MEMS)", proceedings of Modeling and Simulation of Micosystems '01, Hilton Head Island, USA, mars 2001, pp. 169-189, disponible sur <http://www.cr.org>.
- [RENK99] O.Renken and W. Benecke, "Electrostatically driven micromachined nickel resonators and gyroscopes", Proceedings of SPIE, Design, Test, and Microfabrication of MEMS and MOEMS, Volume 3680, 1999, Paris, pp. 716-727.
- [STIN96] Stine, B., D. Boning, J. Chung, D. Bell, E. Equi, "Inter- and Intra-die Polysilicon Critical Dimension Variation", Manufacturing Yield, Reliability, and Failure Analysis session, SPIE 1996 Symposium on Microelectronic Manufacturing, SPIE Vol. 2874, p. 27, Austin TX.

- [STIN97] Stine, B., D. Boning, and J. Chung, "Analysis and Decomposition of Spatial Variation in Integrated Circuit Processes and Devices", IEEE Trans. Semi. Manuf., 1997, 56 p.
- [SWAR98] N. Swart, S. Bart, M. Zaman, et al., "AutoMM: Automatic Generation of Dynamic Macromodels for MEMS Devices", Proceeding of the Eleventh Annual International Workshop on Micro Electro Mechanical Systems, Heidelberg, Germany, 1998, pp. 178-183.
- [WHIT99] J. White, "Fast algorithms for 3-D Simulation", proceedings of Modeling and Simulation of Microsystems 1999, San Juan, USA, disponible sur <http://www.cr.org>.
- [WILS99] N.M. Wilson, S. Liang, P. Pinsky, and R.W. Dutton, "A Novel Method to Utilize Existing TCAD Tools to Build Accurate Geometry Required for MEMS Simulation", technical Proceedings of the 1999 International Conference on Modeling and Simulation of Microsystems, Puerto Rico-U.S.A., 4 p.
- [YE98] W. Ye, S. Mukherjee and N. C. MacDonald, "Optimal Shape Design of an Electrostatic Comb Drive in Micro-Electromechanical Systems", Journal of Micro Electromechanical Systems 1998, pp. 16-27.

# Conclusion et perspectives

## CONCLUSIONS ET PERSPECTIVES

Lors de la fabrication de circuits intégrés, et a fortiori de MEMS, certains paramètres technologiques sont difficilement contrôlables. Pour cette raison, on leur associe une dispersion. Dans certains cas, les circuits ainsi conçus peuvent s'avérer hors spécifications et font chuter le rendement. En modifiant les dimensions des masques, on peut trouver des solutions qui améliorent la robustesse des MEMS : l'incertitude sur les paramètres technologiques est la même, mais ses effets sur les performances sont minimisés. On parle d'optimisation statistique du rendement paramétrique, et c'est le sujet de l'étude présentée dans ce manuscrit.

Dans le premier chapitre, nous avons traité les méthodes de fabrication des MEMS ainsi que les principales applications actuelles. Le thème du rendement en microélectronique et des outils CAO qui y sont associés sont abordés, de même que leur importance pour le monde industriel.

Dans le second chapitre, nous apportons des informations sur les différences fondamentales qui existent entre les paramètres qui ont un rôle à jouer au cours de l'optimisation. Nous passons en revue les principales méthodes d'estimation et d'optimisation du rendement, et revenons de manière plus détaillée sur la méthode des plans d'expériences.

Le troisième chapitre est dédié au nouvel algorithme que nous proposons dans le cadre de l'optimisation du rendement. Il fait appel aux plans d'expériences et se découpe en deux grandes phases : d'abord réduire la variance des performances, ensuite seulement ajuster la valeur moyenne de ces performances. Les choix opérés lors de la mise au point de l'algorithme sont justifiés.

Dans le quatrième et dernier chapitre, la manière dont l'algorithme a été implémenté en langage JAVA est commentée. Des exemples (filtre passe-bande et MEMS) sont donnés et permettent de vérifier le bon fonctionnement de l'algorithme. Les résultats obtenus démontrent bien une amélioration sensible du rendement et certifient donc la bonne tenue de notre méthode.

Toutefois, de nombreux tests et vérifications restent encore à réaliser. Pour s'assurer de sa réelle efficacité (solution optimale trouvée ? Combien de simulations requises ?), il

faudrait réaliser des comparaisons avec les solutions proposées par d'autres logiciels. Malheureusement, nous n'avons pas eu le temps de réaliser assez de tests avec les modules de Cadence (Circuit Surfer) et de Mentor Graphics (Aspire) pour opérer des comparaisons solides (réécriture des modèles, différentes descriptions des objectifs). Nous pouvons ainsi affirmer avoir développé une bonne solution, mais ses avantages sur d'autres restent théoriques tant qu'elles ne seront pas implémentées. Pour pouvoir effectuer de telles comparaisons, il faut réaliser de nombreux tests sur différents circuits, impliquant de nombreux paramètres. De telles contraintes imposent, au vu du nombre de simulations nécessaires, une automatisation du processus, c'est à dire réaliser chaque fois un programme informatique. Cela prendrait trop de temps et n'a pas vraiment sa place dans un travail de thèse qui se place plutôt en aval lors de la synthèse des résultats, ou en amont, lors de la mise au point d'une méthode précise.

En parallèle, des structures de tests (microrésonateurs, etc...) ont été réalisées et certaines encapsulées pour en vérifier le bon fonctionnement. Mais le nombre de telles structures qui doivent être fabriquées pour caractériser statistiquement une technologie est énorme. Cela nécessite un banc de mesure complexe et automatisé, ce qui est encore rare parmi les fondeurs impliqués dans les microsystèmes. Il a donc fallu établir des priorités entre la mise au point d'une méthode d'optimisation du rendement et son application (logiciel), l'implémentation des méthodes classiques afin de les comparer, la caractérisation technologique afin de disposer de données fiables, la modélisation précise d'une structure de test dont la fabrication montrerait la bonne tenue de notre méthode d'optimisation. Nous avons opté pour la mise au point d'un algorithme original d'optimisation et son implémentation, et réalisé un premier run pour caractériser la technologie et mettre au point une structure MEMS intéressante pour le processus d'optimisation.

Réalisée dans le cadre d'une convention CIFRE, les trois années de thèse se sont trouvées fortement imprégnées du milieu industriel du semi-conducteur. La première année s'est passée au LIRMM à Montpellier, et les 2 suivantes à MEMSCAP, à Grenoble. Cette situation m'a permis de voir en peu de temps les principales étapes qui peuvent joncher la vie d'une entreprise (fondée en 1997, entrée en bourse en 2001), de rencontrer beaucoup d'acteurs d'une communauté émergente, celle des microsystèmes, qu'ils soient technologues, concepteurs de MEMS ou développeurs d'outils CAO dédiés. L'outil développé se situe à l'interface de ces trois métiers : c'est un outil de CAO développé pour les besoins des concepteurs de MEMS, en tenant compte des contraintes formulées par les technologues, qui vise à améliorer le rendement de fabrication des MEMS. Modulaire, nous l'avons conçu pour pouvoir l'adapter rapidement à un flot de conception de systèmes

intégrés, quel que soit le simulateur impliqué et le langage de description du système (pour le moment : Eldo et Verilog-A). Actuellement, il ne permet d'optimiser qu'une performance à la fois, et ce en analyse temporelle ou fréquentielle (la partie théorique d'une optimisation multi-performances est traitée dans le chapitre 3). Il permet le suivi de l'évolution des valeurs des paramètres du MEMS en cours d'optimisation et offre une sortie graphique pour l'interprétation des résultats.

# ANNEXES



## ANNEXES

Nous apportons ici quelques précisions sur la nature des principales classes utilisées lors de l'implémentation de l'algorithme d'optimisation statistique. Nous précisons aussi quels fichiers sont utilisés et leur intérêt.

### **I. CLASSES UTILISEES**

- **algo/** regroupe les 4 classes clés de l'algorithme :
  - DoeToSimu réalise la réécriture de la netlist en fonction les plans d'expériences et l'analyse des résultats de la simulation de la netlist.
  - MeanTuning reprend l'algorithme d'optimisation du rendement dans sa phase d'ajustement de la valeur moyenne.
  - SigmaReduction reprend l'algorithme dans sa phase de réduction de la variabilité.
  - YieldEstimation évalue le rendement.
- **app/** regroupe les 2 classes exécutables :
  - StatOpt est l'exécutable du programme.
  - Xcalculator permet de calculer les points d'abscisse utilisés lors de l'affichage des courbes ou de la définition des courbes cibles, préalablement à l'optimisation propre.
- **gui/** regroupe les 3 classes gérant l'interface lors du choix des paramètres à utiliser pour l'optimisation.
- **io/** regroupe les 7 classes dédiées à la lecture où à l'écriture de fichiers ou de grandes quantités de données (résultats de simulations).
  - AlgoData réalise la lecture du fichier où sont saisis les paramètres de l'algorithme (P1, Ps, P2, Pm, boucleAmax, boucleBmax).
  - DataOfSimu enregistre et trie les résultats des simulations.
  - GnuPlot gère la sortie graphique des résultats de l'optimisation : il fait donc des appels à Gnuplot et XView.
  - MatrixReader permet la lecture et l'utilisation des matrices représentant des plans d'expériences qui sont disponibles sous forme de fichiers.
  - NetlistReader met en forme la netlist initiale (suppression des commentaires...) et la réécrit pour un traitement plus simple et rapide. Les lignes sont rangées dans des tableaux (ArrayList en Java).
  - NetlistWriter reprend les tableaux en question pour écrire la netlist.

-PlotWriter enregistre et classe dans des fichiers toutes les courbes obtenues lors des différentes estimations du rendement.

• **db/** regroupe toutes les autres classes (16) :

-AbstractParameter : les paramètres mixtes sont à la fois considérés comme des DesignFactor (facteurs de dessin) et des NoiseFactor (facteurs de bruit). Pour faciliter cette double interprétation, ces deux classes "descendent" (héritent des mêmes propriétés) de la même classe : AbstractParameter.

-Factors, Factor, DesignFactor (associée aux facteurs de dessin), NoiseFactor (associée aux facteurs de bruit), TargetData (description des courbes cibles) sont liées comme on peut le voir sur la Figure 4-8, page 119.

-FactorCreator valide la création d'objets associés aux paramètres choisis pour l'optimisation par l'intermédiaire de l'interface. Cette classe permet aussi d'imposer un nouvel écart type et un nouveau pas (écrasant ceux contenus dans les fichiers de description des paramètres) qui s'appliquent à l'ensemble des paramètres mixtes, calculés d'après un pourcentage de la valeur initiale du paramètre. Cela permet de voir l'influence des paramètres de l'algorithme ( $Ps_1...$ ) et du pas des facteurs vis-à-vis de leur dispersion de manière plus nette, car ils sont ainsi liés par la même relation pour l'ensemble des facteurs. Par exemple, on fixe sigma à 1% de la valeur initiale du paramètre, et le pas à deux sigmas.

-FactorSetter met à jour la valeur d'un paramètre pour la prochaine simulation.

-FixedDesignFactorExperiments permet de traiter les données issues des simulations associées à une combinaison du plan principal.

-Instance : un objet de ce type est associé à chaque instance de la netlist.

-InstanceParameter : un objet de ce type est associé à chaque paramètre d'une instance donnée.

-NetlistInstance permet la lecture de la netlist et la création des objets de type instances nécessaires.

-TechInstance, TechInstanceParameter, TechfileInstance jouent le même rôle que les classes précédentes, mais s'adressent à un éventuel fichier technologique où sont parfois rangés les valeurs des paramètres technologique quand ils ne sont pas dans la netlist même. On utilisera un tel fichier technologique pour une meilleure lisibilité pour cette première implémentation.

-TargetAnalysis : modifie la netlist en fonction de l'analyse requise pour une simulation donnée, et réalise l'appel à Spectre.

## II. FICHIERS UTILISES

Les fichiers utilisés sont placés dans trois répertoires :

- **factors/** Les noms des fichiers décrivent leur contenu :

-D\_nomDel'Instance\_nomDuParamètre (D\_r2\_r) décrit un paramètre de dessin. On y trouve 4 coefficients : nom du paramètre, borne inférieure, borne supérieure, pas.

-N\_nomDel'Instance\_nomDuParamètre (N\_r2\_r) décrit un paramètre de bruit. On y trouve 3 coefficients principaux : nom du paramètre, dispersion, offset éventuel.

-N\_nomDuParamètreTechnologique (N\_YoungModulus) décrit un paramètre technologique, avec le même aspect qu'un fichier de type N\_nomDel'Instance\_nomDuParamètre.

-T\_typeD'analyse\_nœud\_n° (T\_ac\_out\_2) décrit une courbe cible. Il comporte 8 paramètres : le nom de la performance, le type d'analyse qui lui est associée, le nœud où le signal est relevé, le poids de la performance ciblée, les points de début et de fin de l'analyse, le pas utilisé et une précision pour l'affichage : logarithmique ou linéaire. Le type d'affichage, le pas et les bornes de l'analyse permettent de définir les points d'abscisse. En chacun de ces points, on peut définir si nécessaire la valeur ciblée et/ou la valeur minimale acceptée et/ou la valeur maximale acceptée. Ces points pouvant être très nombreux, le programme Xcalculator permet de les générer automatiquement. Il renvoie un fichier `abscisses_file` de la forme suivante :

| abscisse | moyenne | min | max |
|----------|---------|-----|-----|
| x0       | NaN     | NaN | NaN |
| x1       | NaN     | NaN | NaN |
| x2       | NaN     | NaN | NaN |

Ce sont ces lignes que l'on retrouve dans `T_typeD'analyse_nœud_n°`. L'utilisateur remplace juste les "NaN" là où il désire imposer une contrainte.

- DV\_nomDel'Instance\_nomDuParamètre fournit à l'utilisateur l'évolution d'un paramètre de dessin tout au long de l'optimisation

- **inputfiles/** Ce répertoire contient la netlist, le fichier technologique éventuel, et le fichier "algodata" qui contient les paramètres de l'algorithme  $Ps_1$ ,  $Ps_2$ ,  $Ps_3$ ,  $Pe_1$ ,  $Ps_2$ ,  $boucleAmax$ ,  $boucleBmax$  et  $boucleB2max$ .

- **tempfiles/** Ce répertoire héberge tous les autres fichiers manipulés :

-DFDOEMonitoring : retrace les valeurs prises par les paramètres de dessin dans le plan principal.

-NFDOEMonitoring : retrace les valeurs prises par les paramètres de bruit dans le plan bruit, pour chaque combinaison du plan principal.

-Targets : définit quelles sont les performances considérées pour l'optimisation.

-abscisses\_file contient les abscisses des points intéressants, soit pour le processus d'optimisation, soit simplement pour l'affichage des courbes.

-T\_typeD'analyse\_nœud\_n°\_graph\_sigeps : graphe retraçant l'évolution de l'erreur sur la valeur moyenne d'une performance ainsi que sa dispersion avant et après optimisation. Ce fichier est généré par la classe GnuPlot.

-T\_typeD'analyse\_nœud\_n°\_graph\_simus : courbes de l'ensemble des simulations réalisées lors des évaluations du rendement.

-T\_typeD'analyse\_nœud\_n°\_simus.gnu contient le script généré par la classe GnuPlot pour lancer le programme Gnuplot.

-T\_typeD'analyse\_nœud\_n°\_ye\_n contient les coordonnées des points des courbes obtenues pour la performance T\_typeD'analyse\_nœud\_n° lors de la n<sup>ième</sup> estimation du rendement.

-T\_typeD'analyse\_nœud\_n°\_target contient les coordonnées des contraintes s'appliquant à la performant nommée T\_typeD'analyse\_nœud\_n°.

-writtenetlist contient la netlist réécrite sans commentaires.

-writtenetlist\_typeD'analyse contient la netlist ou une analyse a été spécifiée.

-writtenetlist\_typeD'analyse.dat contient les résultats de la simulation de cette netlist.

-netlist\_sigma\_opt est la netlist à l'issue de la réduction de la variance.

-netlist\_mean\_opt est la netlist à l'issue de la réduction de la fonction erreur associée aux spécifications.

-netlist\_mean2\_opt est la netlist à l'issue de l'éventuelle demande d'une réduction de la fonction erreur avec un pas plus fin.

### III. LES FILTRES A PEIGNES ELECTROSTATIQUES

Cette structure se retrouve très souvent dans la littérature. Elle se décline sous plusieurs formes et permet notamment la réalisation de filtres électro-mécaniques dont le principal inconvénient jusqu'à maintenant est une fréquence de résonance propre assez basse qui ne permet pas de l'utiliser pour des applications RF. Cette fréquence de résonance mécanique est fixée par la géométrie et les matériaux de la structure.

#### • Filtres MEMS

Le premier du genre est aussi souvent décrit comme le premier MEMS [Figure 0-1], réalisé en 1967 [NATH67-page 40].

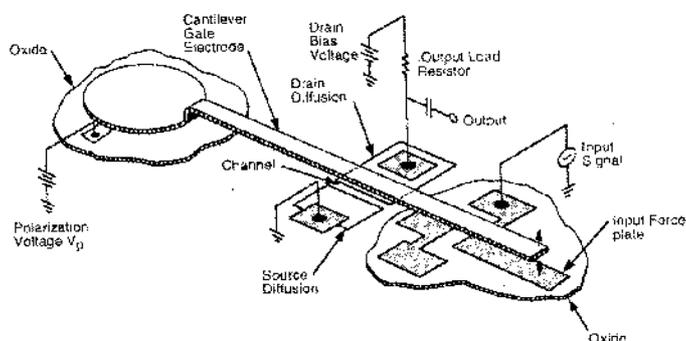


Figure 0-1 : Resonant Gate Transistor

Nguyen a décrit un filtre ayant une fréquence de résonance très élevée (quelques dizaines de MHz) [Figure 0-2], mais il atteint de telles performances sous vide et au prix d'une tension d'actuation de plusieurs centaines de volts [NGUY99-p. 145].

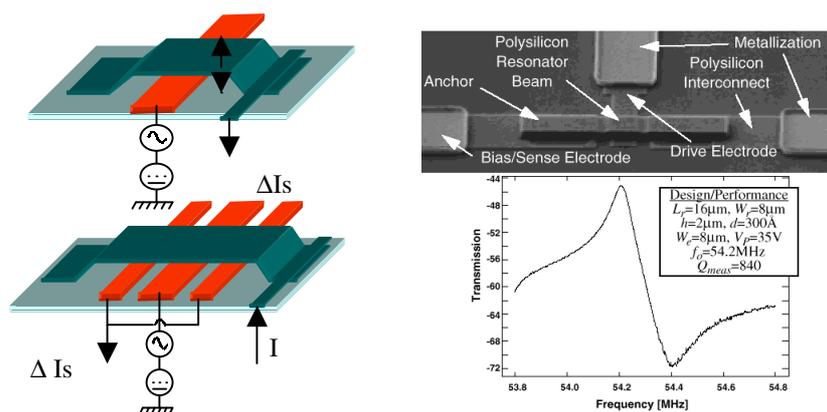
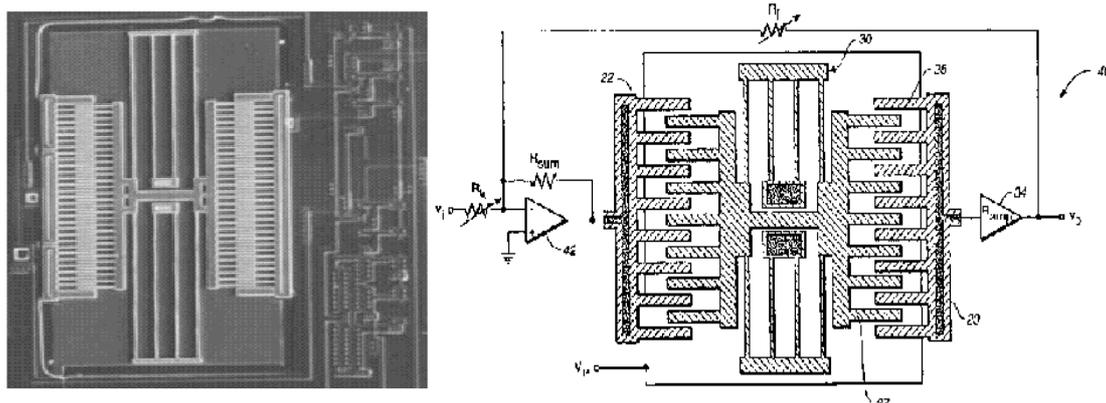


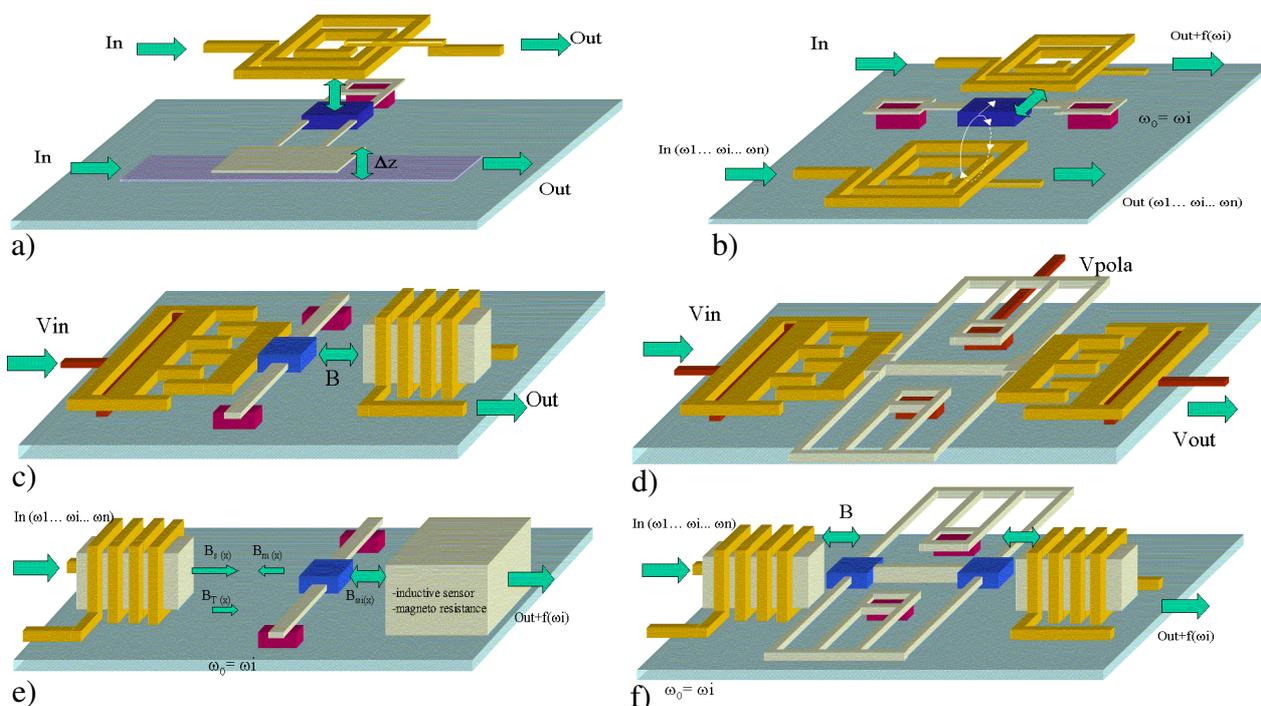
Figure 0-2 : Principe des structures de Nguyen résonnant verticalement

Nous nous sommes inspiré de la version plus traditionnelle (filtre à peignes latéraux) [Figure 0-3] : la fréquence de résonance n'est que de quelques dizaines de kHz, mais la tension d'actuation requise était bien adaptée, et la tension d'actuation requise raisonnable : 10V dans l'air ambiant.



**Figure 0-3 : Structure résonante à peignes électrostatique latéraux [NGUY98]**

Plusieurs structures ont toutefois été envisagées [Figure 0-4]. On pouvait effectivement imaginer combiner de différentes manières des modules mécaniques, électrostatiques ou magnétiques, pour atteindre le même but.



**Figure 0-4 : Variantes de filtres MEMS**

C'est donc la forme d) qui a été retenue. Les déplacements prévus ont été vérifiés, mais pas les fréquences propres d'oscillation. Une caractérisation visuelle simple n'a pu être montée sans système stroboscopique, et une caractérisation électronique aurait nécessité la mise au point d'un circuit permettant de mesurer des variations de capacités de l'ordre de quelques femto Farads, trop longue à réaliser.

#### **IV. RESSOURCES**

Internet est une source importante d'information. Concernant les MEMS, les sites de Sandia, de la DARPA et MemsNet sont particulièrement riches.

DARPA [http://www.darpa.mil/mto/index\\_b.html](http://www.darpa.mil/mto/index_b.html)

SANDIA <http://www.mdl.sandia.gov/>

MEMS Clearing House <http://www.memsnet.org/>

Il existe aussi quelques journaux qui sont mis en lignes, ainsi que des proceedings de conférences :

Journal of Microlithography, Microfabrication, and Microsystems (JM3) :

<http://ojs.aip.org/jmm/index.jsp>

Sensors and Actuators : <http://www.elsevier.com/homepage/saa/sensors/>

Sensorsmag <http://www.sensorsmag.com/>

Archives de Modeling and Simulation of Microsystems : <http://www.cr.org/>

Journal of Micromechanics and Microengineering :

<http://www.iop.org/EJ/S/UNREG/journal/0960-1317>

Journal of Micromechatronics : <http://www.vspub.com/journals/jn-JouMic.html>

Archives de Design Automation Conference :

<http://www.sigda.org/Archives/ProceedingArchives/Dac/>

Journal of MEMS (ieee-asme) :

<http://www.asmeny.org/Catalog/Html/catqper.htm>

<http://shop.ieee.org/store/Overviews/periodicals.asp#list>

Concernant les conférences, ces dernières n'ont souvent pas de site attiré. Les principales sont référencées avec les dates importantes sur les sites suivants :

<http://mmc.mpe.ntu.edu.sg/links.asp>

<http://home.earthlink.net/~trimmerw/mems/Conferences.html>

<http://www.smalltimes.com/calendar/index.cfm>

<http://www.memsnet.org/events/>

<http://www.mic.dtu.dk/research/mems/conferences.htm>

**V. PUBLICATIONS**

Flavien Delauche, B. Affour, C. Dufaza, "Parametric yield optimization of a micro-resonator using ASPIRE", dans "Reliability, Testing, and Characterization of MEMS/MOEMS", R. Ramesham (ed), vol. 4558, Proceedings of SPIE Conference on Micromachining and Microfabrication 2001, San Francisco-USA,, p. 22-31.

Flavien Delauche, B. Affour, C. Dufaza, "Parametric yield optimization of MEMS", Proceedings of DTIP'2002, Symposium on Design, Test, Integration and Packaging of MEMS/MOEMS", Cannes, p. 126-135.



## RESUME

Les microsystèmes ou MEMS (Micro Electro Mechanical Systems) entrent dans une phase industrielle. Habituellement conçus à partir de procédés utilisés classiquement en microélectronique, ils peuvent aussi inclure des parties non électroniques : mécaniques, optiques, etc. Leur complexité technologique et la difficulté à modéliser ces applications multi-physiques ont retardé leur industrialisation, mais cette phase est aujourd'hui atteinte. En parallèle se développent donc des contraintes de coût de fabrication et de rendement dans le milieu très concurrentiel du semiconducteur. Les travaux réalisés dans le cadre de cette thèse visent à développer un logiciel implémentant un algorithme original d'optimisation statistique du rendement paramétrique qui tente de s'affranchir des principales limites associées aux méthodes courantes. Il se présente comme un module aisément insérable dans un flot de conception de circuit intégré ou de MEMS. Il s'agit d'optimiser les dimensions des structures réalisées afin de limiter les variations et dégradations des performances des microsystèmes qui sont dues aux fluctuations inévitables des paramètres technologiques lors de leur fabrication, et sont causes de mauvais rendement.

Mots clés : Rendement paramétrique, Optimisation statistique, Plans d'expériences, Microsystèmes, Microrésonateur, Réduction de la variance, Méthode Taguchi.

## ABSTRACT

Statistical Optimization of Parametric Yield for MEMS

MEMS are now entering an industrial era. Fabrication processes are usually based on integrated circuits ones, but they can also include non-electronical parts such as mechanical or optical ones. As multiphysics applications, MEMS appear to be very complex and hard to be accurately modeled, which lead their industrialization to be delayed until today. Now, they have to face the aggressivity of semiconductor market, and yield management is a critical success factor for this new technology. In this dissertation, we detail the development of a software for MEMS parametric yield statistical optimization which can be easily included in integrated circuits or MEMS conception flow. By optimizing layout parameters, it allows to reduce MEMS performances variations and dispersion which are due to unavoidable technological parameters fluctuation during the different process steps, and lead to yield losses.

Key words : Parametric yield management, Statistical optimization, Design Of Experiments, MEMS, Microresonator, Variability minimization, Design For Manufacturability, Taguchi method.

## DISCIPLINE

micro-électronique

## STRUCTURES D'ACCUEIL

MEMSCAP S.A.  
Parc des Fontaines  
38926 BERNIN

LIRMM  
Université Montpellier II / CNRS  
161 rue Ada  
34392 Montpellier cedex 5