

A Toufeek, mon grand père...

A Victoria, ma mère...

A Hussein, mon père

A May, ma femme et Hussein mon fils.

*A mes frères et soeurs Marwa, Nora, Mohammad, Youssra, Anas,
Ali, Ahmad, Sam, et Boushra.*

*A mes neveux, et mes nièces : Ali, Johanna, Dima, Hussein,
Alizen et Victoria.*

A Oussama, Adnan, Ali, Firas, et Rim.

REMERCIEMENTS

Je tiens à exprimer ma reconnaissance à...

Monsieur Pierre Gentil, pour l'honneur qu'il me fait en acceptant de présider le jury de cette thèse.

Madame Marta Rencz, pour avoir accepté d'être rapporteur de ce travail, et de s'être déplacée de si loin pour la soutenance.

Monsieur Christian Dufaza pour avoir accepté d'être rapporteur de ce travail, et les nombreuses corrections qui ont amélioré le manuscrit.

Monsieur Bernard Courtois, pour m'avoir accueilli au sein du groupe microsystemes, et pour m'avoir donné l'opportunité d'entreprendre de nombreux projets.

Monsieur Kholdoun Torki, pour m'avoir encadré et pour toute l'aide qu'il m'a fourni tout au long de cette thèse.

Mes collègues et amis au laboratoire TIMA, dont la liste est longue pour tenir dans une page.

Mes compatriotes en France qui m'ont soutenu pendant les moments difficiles.

Au gouvernement syrien qui a financé mes études en France.

Résumé

Outils CAO pour Microsystèmes

On peut définir les microsystèmes ou les systèmes micro-électromécaniques comme des systèmes de petites dimensions (plus petites qu'un centimètre cube) et qui accomplissent des fonctions de précision. Les microsystèmes se composent de capteurs pour acquérir les informations du monde extérieur, d'une partie électronique pour le traitement de données et d'actionneurs qui réagissent avec le monde extérieur. Le coût élevé de développement d'un microsystème est dû en grande partie à la complexité du flot de conception faisant intervenir une multitude d'outils CAO spécifiques ou adaptés aux microsystèmes.

Le but de cette thèse est de développer une méthodologie de modélisation et de simulation de microsystèmes en se basant sur les outils CAO microélectronique existants. Cette méthodologie comprend les différents niveaux d'abstraction des microsystèmes. Pour transposer les techniques de simulation et modélisation du domaine microélectronique aux microsystèmes, il a fallu faire tout d'abord une étude comparative entre les deux flots de conception. Nous avons ensuite développé des techniques pour gérer la complexité des dessins du masque de microsystèmes. Ces techniques couvrent la génération des motifs uniformes au niveau de layout, ainsi que l'adaptation des algorithmes de vérification de règles de dessin aux formes complexes microsystèmes. Une étude sur une méthodologie de synthèse et d'optimisation pour microsystèmes est présentée où un langage hybride de description de microsystèmes est illustré et différents algorithmes d'optimisation sont discutés. Finalement un environnement de simulation globale de SoC est défini où un composant microsystème « commutateur optique » est simulé avec son environnement électronique.

Mots clés : Microsystèmes, MEMS, CAO, Modélisation, Simulation, Optimisation, Synthèse, Co-simulation, Formes Non Manhattan, SoC.

Abstract

CAD tools for MEMS

With the rapid development of MicroElectroMechanical Systems "MEMS" and to reduce their high prototyping costs, there is a rising need for Computer Aided Design "CAD" tools which are able to handle the MEMS devices during their different steps of design and fabrication. A common approach to MEMS CAD tools is to make use of integrated circuits CAD tools by adding specific enhancements for MEMS designs.

The aim of this thesis is to define a modeling and simulation methodology for MEMS which covers the different abstraction levels. We started our study by comparing the design flow of microelectronics and MEMS, then we developed a new methodology to handle MEMS layout. This methodology covers the generation of complex MEMS design shapes (especially Non-Manhattan), and the adjustment of the design rule checking "DRC" algorithms to be able to handle these complex shapes. A synthesis and optimization methodology is presented. Through this methodology we are developing a MEMS description language and discussing different optimization algorithms. Finally we are presenting a methodology to establish an environment for the global modeling and simulation of systems on chip "SoC" embedding MEMS devices. In this environment a multi-language/multi-engine approach is used at high-level design description. This methodology is proved through an optical switch co-simulation.

Keywords: MEMS, CAD, Modeling, Simulation, Co-simulation, Synthesis, Non-Manhattan shapes, SoC.

Table des matières

1	Introduction	1
1.1	Etat de l'art	4
1.2	Présentation de la thèse	6
2	Flot de conception microsysteme	11
2.1	Structure globale d'un environnement de conception microsysteme	12
2.2	Flot de conception microélectronique	13
2.3	Flot de conception microsysteme	14
2.4	Le flot de conception microsysteme face au flot de conception microélectronique	17
3	Technologies de fabrication et outils de simulation de procédé	21
3.1	Les technologies microsystemes	21
3.1.1	Le micro-usinage en surface	21
3.1.1.1	La micromécanique	22
3.1.1.2	Les principaux fondeurs multi-utilisateurs	22
3.1.1.3	La planarisation mécano chimique	23
3.1.1.4	Les problèmes de contact de surface et les dimples	24
3.1.1.5	Les technologies de micro-usinage de surface intégrées	24
3.1.2	Le micro-usinage en volume	27
3.1.2.1	La gravure anisotrope du silicium	28
3.1.2.2	Intérêt de la gravure anisotrope	28
3.1.2.3	L'anisotropie de la réaction de gravure	29
3.1.2.4	Les solutions de gravure anisotrope du silicium	30
3.1.2.5	Les différents types de micro-usinage en volume	30
3.1.2.6	Le micro-usinage en volume face arrière	31
3.1.2.7	Le micro-usinage en volume face avant et arrière	32
3.1.2.8	Le micro-usinage en volume électrochimique	32
3.1.3	D'autres technologies	33
3.1.3.1	Les microsystemes sur SOI	33
3.1.3.2	Le LIGA	34
3.2	Simulation de procédé technologique et génération de modèle 3D d'une structure microsysteme	35
3.2.1	Définition du procédé	36
3.2.1.1	Déposition d'une couche	36
I.	Déposition « Conformal » :	36
II.	Déposition « Snowfall »	37
III.	Déposition de remplissage « Fill »	37
3.2.1.2	Suppression d'une couche (gravure)	38
I.	Gravure en surface	38
II.	Gravure en volume	39
i	La théorie géométrique de la gravure bidimensionnelle	40
ii	L'algorithme de simulation de la gravure bidimensionnelle	42
iii	Implémentation, simulation et résultats	45

3.2.2	Construction du modèle 3D	46
3.3	Conclusion :	52
4	Modélisation composant et système	55
4.1	Modélisation par description analytique puis calcul numérique	55
4.1.1	Constantes localisées réseaux de Kirchoff	55
4.1.1.1	Introduction	55
4.1.1.2	Étude de transducteurs élémentaires	57
I.	Introduction	57
II.	Configurations de base	58
III.	Étude de transducteurs électrostatiques	58
IV.	Équations caractéristiques	61
4.2	Modélisation par éléments finis	63
4.3	La simulation comportementale des microsystemes	64
4.3.1	Simulateurs de circuits	64
4.3.2	La modélisation par langage comportemental	65
4.3.2.1	Structure d'un modèle HDL	66
4.3.2.2	Exemple électrothermique convertisseur électrothermique	66
•	Analogie électrique/thermique	67
4.3.2.3	Exemple électromécanique : les micro-résonateurs électrostatiques en peigne	68
•	Description du composant	68
•	Formulation mathématique	69
•	Analogie Electrique Mécanique	71
4.4	Simulation globale de microsystemes	72
4.5	Conclusion	75
5	Dessin du masque et règles de dessin pour microsystemes	79
5.1	Introduction	79
5.2	Les Formes Non-Manhattan	79
5.3	La discrétisation des formes Non-Manhattan	81
5.3.1	Fondements mathématiques	82
5.3.1.1	Discrétisation d'une courbe	82
I.	Discrétisation par pavés semi-ouvert	82
II.	Discrétisation par intersection avec la grille	82
III.	Discrétisation par intersection avec la grille d'un même côté de la courbe	83
5.3.1.2	Discrétisation d'un segment de droite	83
I.	Plus proche voisin discret	83
II.	Plus proche voisin discret de même côté	83
III.	Plus proche voisin selon une direction de grille	84
5.3.1.3	Discrétisation des formes coniques	85
5.3.1.4	Discrétisation des polygones avec l'angle libre (Non-Manhattan)	86
5.4	Les règles de dessin pour microsystemes	87
5.4.1	Les fausses erreurs d'origine topologique	88
5.4.2	Technique de résolution des problèmes topologiques	90
5.4.3	Les fausses erreurs d'origine Non-Manhattan	91
5.4.4	Technique pour résoudre les problèmes de fausse signalisation d'origine Non-Manhattan	92
5.5	Conclusions	93
6	Techniques de synthèse et d'optimisation de microsystemes	96
6.1	Introduction	96

6.2	Méthodologie de synthèse pour microsysteme	98
6.3	Technique d'optimisation	103
6.3.1	Processus d'optimisation	104
6.4	Optimisation de microsysteme : optimisation d'un convertisseur électrothermique	106
6.4.1	Introduction	106
6.4.2	Expérimentation et résultats	108
6.4.2.1	Premier mode : la constante de temps τ comme fonction objective	111
I.	Première méthode : algorithme du gradient de la plus forte pente descendante	111
II.	Deuxième méthode : l'algorithme BFGS, $H_0=I$	113
III.	Comparaison des deux méthodes	115
6.4.2.2	Deuxième mode : la réponse R comme fonction objective	115
I.	Première méthode : algorithme du gradient de la plus forte pente descendante	116
II.	Deuxième méthode : l'algorithme BFGS, $H_0=I$	118
III.	Troisième méthode : l'algorithme BFGS, $H_0 = \nabla^2 f(\mathbf{x}_0)$	120
IV.	Comparaison des trois méthodes	121
6.4.2.3	Troisième mode : la surface S comme fonction objective	122
I.	Première méthode : algorithme du gradient de la plus forte pente descendante	123
II.	Deuxième méthode : l'algorithme BFGS, $H_0=I$	125
III.	Troisième méthode : l'algorithme BFGS, $H_0 = \nabla^2 f(\mathbf{x}_0)$	128
IV.	Comparaison des deux méthodes	131
6.4.2.4	Comparaison entre les trois modes d'optimisation	131
6.5	Conclusion	132
7	Conclusions et perspectives	135
	Publications	137
	Annexe A : Modèle HDL-A d'un convertisseur électrothermique	138
	Annexe B: Bosch technology design rules	141
	Annexe C: AMS technology design rules	149
	Annexe D : Méthodes et algorithmes d'optimisation	163

Liste des figures

Figure 1.1. Représentation schématique d'un microsysteme qui montre l'interaction entre les différents composants (capteurs, actionneurs et l'électronique nécessaire) et l'environnement. Différents niveaux de modélisation et simulation sont aussi montrés. _____	2
Figure 2.1. Structure globale d'un environnement de conception microsysteme _____	13
Figure 2.2. Flot de conception Microélectronique. _____	14
Figure 2.3. Flot de conception des microsystemes. _____	15
Figure 3.1. Schéma en coupe d'une technologie de micro-usinage en surface avant (a) et après (b) gravure des oxydes sacrificiels. _____	22
Figure 3.2. Photos MEB d'éléments micromécaniques fabriqués par micro-usinage de surface (technologie MUMPS), charnière (a), dents d'engrenage (b), ressort pivotant (c). _____	22
Figure 3.3. Photo MEB d'un micromoteur électrostatique fabriqué en technologie MUMPS. _____	23
Figure 3.4. Photos MEB de structures micro-usinées en surface avec planarisation (a) (Courtesy, Sandia National Labs - MEMS, S&T Department, www.mems.sandia.gov) et sans planarisation (b). _____	24
Figure 3.5. Photo MEB d'un dimple sous une couche de polysilicium suspendue (Courtesy, Sandia National Labs - MEMS, S&T Department, www.mems.sandia.gov). _____	24
Figure 3.6. Schéma en coupe d'une technologie de micro-usinage en surface intégrée [19]. _____	25
Figure 3.7. Schéma en coupe d'une technologie MEMS intégrée avec parties microsystemes enterrées, (Courtesy, Sandia National Labs - MEMS, S&T Department, www.mems.sandia.gov). _____	26
Figure 3.8. Schéma en coupe d'une technologie microsysteme intégrée avec parties déposées et gravées après l'électronique et l'usage de niveaux métalliques en tungstène. _____	26
Figure 3.9. Schéma en coupe d'une technologie de micro-usinage en volume face avant compatible CMOS. _____	27
Figure 3.10. Image MEB d'une microstructure obtenue par micro-usinage en volume face avant et montrant la composition en sandwich des structures suspendues. _____	28
Figure 3.11. Diagramme polaire des vitesses de gravure du TMAH. _____	28
Figure 3.12. Schéma représentant l'avancée de la gravure anisotrope à partir d'ouvertures triangulaires. _____	29
Figure 3.13. Photo MEB d'une microstructure en pont obtenu par micro-usinage en volume par gravure anisotrope du silicium du substrat. _____	29
Figure 3.14. Schéma des différentes microstructures que l'on peut obtenir par micro-usinage en volume face avant. _____	31
Figure 3.15. Schéma d'une structure à masse suspendue obtenue en micro-usinage en volume face avant [19]. _____	31
Figure 3.16. Schéma des différentes structures résultantes d'un micro-usinage en volume face arrière. _____	32
Figure 3.17. Schéma en coupe d'une structure à masse suspendue obtenue en micro-usinage en face avant et en face arrière [19]. _____	32
Figure 3.18. Schéma en coupe de structures suspendues par micro-usinage électrochimique en volume, face arrière (a) et face avant (b). _____	33
Figure 3.19. Schéma en coupe de la technologie de micro-usinage sur SOI : Epi-SOI de Tronic's microsystems (Courtesy of TRONIC'S Microsystems, France). _____	33
Figure 3.20. Procédé LIGA pour la fabrication de pièces métalliques ou de moules métalliques pour injection de plastiques [19]. _____	34
Figure 3.21. Le flot dans un environnement de modélisation 3D _____	35
Figure 3.22. Déposition « conformal » d'une couche sur une autre. _____	36
Figure 3.23. Déposition sur les faces ombrées. _____	37
Figure 3.24. Déposition « Snowfall ». _____	37
Figure 3.25. Déposition « Fill ». _____	38
Figure 3.26. Gravure en surface inclusive. _____	38
Figure 3.27. Gravure en surface exclusive. _____	39
Figure 3.28. Les paramètres de gravure en surface. _____	39
Figure 3.29. Simulation atomistique de la gravure anisotrope d'une membrane soutenue par quatre bras à 45°, pour différents pas de calculs montrant la progression de la gravure. _____	40
Figure 3.30. Apparition d'une nouvelle face. _____	41
Figure 3.31. Disparition d'une face. _____	42
Figure 3.32. Intersection de deux faces parallèles. _____	42

Figure 3.33. <i>Intersection des faces non parallèles.</i>	42
Figure 3.34. <i>Les différentes configurations de vitesses.</i>	43
Figure 3.35. <i>Simulation de gravure d'une poutre avec 2 pas différents de temps.</i>	45
Figure 3.36. <i>Simulation géométrique (a) de la gravure anisotrope d'une membrane soutenue par quatre bras à 45°, et photo MEB (b) de la même structure avec gravure TMAH incomplète.</i>	45
Figure 3.37. <i>Processus technologique de 4 étapes :(1) déposition Snowfall, (2) déposition Fill, (3) déposition du masque de gravure, (4) gravure inclusive en surface.</i>	47
Figure 3.38. <i>Construction d'un volume 3D à partir d'un polygone 2D.</i>	49
Figure 3.39. <i>Déposition Snowfall de couche L2 sur une couche L1.</i>	49
Figure 3.40. <i>Déposition Conformal de couche L2 sur une couche L1.</i>	50
Figure 3.41. <i>Déposition " Conformal " de deux couches et une vue "cross section" de la même structure.</i>	51
Figure 3.42. <i>Génération d'une poutre par une gravure anisotrope.</i>	51
Figure 3.43. <i>Génération d'une bobine par une gravure en surface.</i>	51
Figures 4.1. <i>Un transducteur à un seul degré de liberté modélisé par les éléments à constantes localisées.</i>	56
Figure 4.2. <i>Un transducteur électromécanique représenté comme quadripôle</i>	57
Figure 4.3. <i>Les différentes catégories de transducteurs électromécaniques.</i>	58
Figure 4.4. <i>Un modèle modifié d'un transducteur électromécanique pour résoudre le problème d'instabilité.</i>	60
Figure 4.5. <i>Exemples d'éléments assemblés en primitives géométriques.</i>	63
Figure 4.6. <i>Différents types de maillage : avec éléments cubiques (a), avec éléments tétraédriques(b), avec maillage adaptatif (c) et avec maillage plus fin (d).</i>	64
Figure 4.7. <i>Schéma (a) et image MEB (b) d'un convertisseur électrothermique à poutre suspendue.</i>	67
Figure 4.8. <i>Modélisation en équivalent électrique d'un convertisseur électrothermique.</i>	67
Figure 4.9. <i>Schéma de principe du fonctionnement d'un micro-résonateur en peigne.</i>	68
Figure 4.10. <i>Connexions électriques d'un micro-résonateur en peigne.</i>	68
Figure 4.11. <i>Image MEB de micro-résonateurs en peigne, à déplacement longitudinal.</i>	69
Figure 4.12. <i>Environnement de co-simulation.</i>	72
Figure 4.13. <i>Multiplexeur optique à micro-miroirs.</i>	73
Figure 4.14. <i>Architecture d'un multiplexeur à micro-miroir</i>	74
Figure 4.15. <i>Architecture d'un multiplexeur à micro-miroir.</i>	74
Figure 4.16. <i>Co-Simulation de commutation</i>	75
Figure 5.1. <i>Les formes Manhattan</i>	80
Figure 5.2. <i>Les formes Non-Manhattan.</i>	80
Figure 5.3. <i>Echantillonnage d'une spirale en polygones.</i>	81
Figure 5.4. <i>(a)Echantillonnage hors grille, (b) Echantillonnage non-uniforme.</i>	81
Figure 5.5. <i>Discrétisation par pavés semi-ouverts.</i>	82
Figure 5.6. <i>Discrétisation par intersection avec la grille, montrant les points supprimés par rapport au résultat de la Figure 5.2.</i>	82
Figure 5.7. <i>Discrétisation par intersection avec la grille d'un même côté de la courbe.</i>	83
Figure 5.8. <i>Discrétisation d'un segment par le plus proche voisin discret.</i>	83
Figure 5.9. <i>Discrétisation d'un segment par le plus proche voisin discret de même côté.</i>	84
Figure 5.10. <i>Discrétisation d'un segment par le plus proche voisin selon la direction de la grille.</i>	84
Figure 5.11. <i>Segment de droite discrète et caractérisation syntaxique.</i>	85
Figure 5.12. <i>Discrétisation de forme conique : cas initial.</i>	85
Figure 5.13. <i>Discrétisation d'un segment dont la pente est <1 et >-1.</i>	86
Figure 5.14. <i>Discrétisation d'un segment dont la pente est >1 ou >-1.</i>	87
Figure 5.15. <i>Les fausses et les vraies signalisations.</i>	88
Figure 5.16. <i>Règles microsystemes dans la technologie AMS.</i>	89
Figure 5.17. <i>Règles microsystemes dans la technologie SensoNor.</i>	89
Figure 5.18. <i>Les règles fonctionnelles dans la technologie Bosch.</i>	90
Figure 5.19. <i>Structure hiérarchique de règles de dessin pour microsystemes.</i>	90
Figure 5.20. <i>Zone microsysteme et zone microélectronique.</i>	91
Figure 5.21. <i>Résultats de la vérification de règles de dessin classique sur une spirale.</i>	92
Figure 5.22. <i>Résultats de vérification de règles de dessins : (a) exécution d'un algorithme classique, (b) exécution d'un algorithme avec la tolérance.</i>	93
Figure 6.1. <i>Schéma simplifié de cycle de conception microsysteme.</i>	96
Figure 6.2. <i>Environnement de synthèse pour microsysteme.</i>	98
Figure 6.3. <i>Paramètres de conception sur une structure de convertisseur électrothermique.</i>	99
Figure 6.4. <i>Décomposition d'un convertisseur électrothermique dans des éléments de base.</i>	100
Figure 6.5. <i>Structure d'un thermocouple</i>	100

Figure 6.6. Entrée/sortie d'un thermocouple. _____	100
Figure 6.7. Résistance de chauffe. _____	101
Figure 6.8. Entrée/sortie d'un thermocouple. _____	102
Figure 6.9. La poutre _____	102
Figure 6.10. L'architecture d'un convertisseur électrothermique. _____	103
Figure 6.11. Processus d'optimisation. _____	105
Figure 6.12. Paramètres R et τ de la réponse du convertisseur électrothermique. _____	106
Figure 6.13. Étapes de l'optimisation d'un convertisseur électrothermique. _____	110
Figure 6.14. Graphe de convergence en utilisant l'algorithme de la plus forte pente descendante, et les valeurs finales de l'optimisation (τ comme fonction objective principale). _____	112
Figure 6.15. Résultats de simulation du comportement du convertisseur avant l'optimisation : (courbe $v(2)$) et après l'optimisation : (courbe $v(3)$) (τ comme fonction objective principale, l'algorithme de la plus forte pente descendant) et valeurs finales des fonctions objectives. _____	113
Figure 6.16. Graphe de convergence en utilisant la méthode BFGS, $H_0=I$, et valeurs finales de l'optimisation (τ comme fonction objective principale). _____	114
Figure 6.17. Résultats de simulation du comportement du convertisseur avant l'optimisation : (courbe $v(2)$) et après l'optimisation : (courbe $v(3)$) (τ comme fonction objective principale, l'algorithme BFGS $H_0=I$ comme méthode d'optimisation), et valeurs finales des fonctions objectives. _____	114
Figure 6.18. Tableau comparatif des résultats de simulation obtenus par les deux méthodes et les valeurs correspondantes des trois fonctions objectives. _____	115
Figure 6.19. Graphe de convergence en utilisant l'algorithme de la plus forte pente descendante, et valeurs finales de l'optimisation (R comme fonction objective principale). _____	117
Figure 6.20. Résultats de simulation du comportement du convertisseur avant l'optimisation: (courbe $v(2)$) et après l'optimisation: (courbe $v(3)$) (R comme fonction objective principale, l'algorithme de la plus forte pente descendant), les valeurs finales des fonctions objectives. _____	118
Figure 6.21. Graphe de convergence en utilisant la méthode BFGS, $H_0=I$, valeurs finales de l'optimisation (R comme fonction objective principale). _____	119
Figure 6.22. Résultats de simulation du comportement du convertisseur avant l'optimisation: (courbe $v(2)$) et après l'optimisation: (courbe $v(3)$) (R comme fonction objective principale, l'algorithme BFGS $H_0=I$), et valeurs finales des fonctions objectives. _____	119
Figure 6.23. Graphe de convergence utilisant la méthode BFGS, $H_0 = \nabla^2 f(x_0)$ (graphe à gauche), valeurs finales de l'optimisation, R comme fonction objective principale. _____	120
Figure 6.24. Résultats de simulation du comportement du convertisseur avant l'optimisation: (courbe $v(2)$) et après l'optimisation: (courbe $v(3)$) (R comme fonction objective principale, l'algorithme BFGS, $H_0 = \nabla^2 f(x_0)$), et valeurs finales des fonctions objectives. _____	121
Figure 6.25. Tableau comparatif des résultats de simulation obtenus par les trois méthodes et les valeurs correspondantes des trois fonctions objectives. _____	122
Figure 6.26. Résultats obtenus par la méthode de la plus forte pente descendante pour le <u>premier</u> point de départ où il n'y a pas de convergence (graphe en haut à gauche) et un zoom sur la zone de divergence montrant le phénomène de zigzag (en haut à droite). _____	123
Figure 6.27. Résultats obtenus par la méthode de la plus forte pente descendante pour le <u>deuxième</u> point de départ où il n'y a pas de convergence (graphe en haut à gauche) et un zoom sur la zone de divergence montrant le phénomène de zigzag (en haut à droite). _____	124
Figure 6.28. Graphe de convergence obtenu en utilisant la méthode BFGS, $H_0=I$ et en partant du <u>premier</u> point de départ, un zoom sur la zone de convergence, et valeurs finales de l'optimisation (S comme fonction objective principale). _____	125
Figure 6.29. Résultats de simulation du comportement du convertisseur avant l'optimisation : (courbe $v(2)$) et après l'optimisation : (courbe $v(3)$) (S comme fonction objective principale, l'algorithme BFGS $H_0=I$, <u>premier</u> point de départ), et valeurs finales des fonctions objectives. _____	126
Figure 6.30. Graphe de convergence en utilisant la méthode BFGS, $H_0=I$ et en partant du <u>deuxième</u> point de départ, et valeurs finales de l'optimisation (S comme fonction objective principale). _____	127
Figure 6.31. Résultats de simulation du comportement du convertisseur avant l'optimisation : (courbe $v(2)$) et après l'optimisation : (courbe $v(3)$) (S comme fonction objective principale, l'algorithme BFGS $H_0=I$, <u>deuxième</u> point de départ), et valeurs finales des fonctions objectives. _____	127
Figure 6.32. Graphe de convergence en utilisant la méthode BFGS, $H_0 = \nabla^2 f(x_0)$ et en partant du <u>premier</u> point de départ, et valeurs finales de l'optimisation (S comme fonction objective principale). _____	128

-
- Figure 6.33.** Résultats de simulation du comportement du convertisseur avant l'optimisation : (courbe $v(2)$) et après l'optimisation : (courbe $v(3)$) (S comme fonction objective principale, l'algorithme BFGS, $H_0 = \nabla^2 f(x_0)$, premier point de départ), et valeurs finales des fonctions objectives . _____ 129
- Figure 6.34.** Graphe de convergence en utilisant la méthode BFGS, $H_0 = \nabla^2 f(x_0)$ et en partant du deuxième point de départ, et valeurs finales de l'optimisation (S comme fonction objective principale). _____ 130
- Figure 6.35.** Résultats de simulation du comportement du convertisseur avant l'optimisation : (courbe $v(2)$) et après l'optimisation : (courbe $v(3)$) (S comme fonction objective principale, l'algorithme BFGS, $H_0 = \nabla^2 f(x_0)$, deuxième point de départ), et valeurs finales des trois fonctions objectives. _____ 131
-

Liste des tableaux

Tableau 4.1. Les matrices $[B]$ et $[T]$ pour les quatre configurations possibles de transducteurs _____	62
Tableau 4.2. Les coefficients couplage et de transduction pour les quatre configurations possibles de transducteurs _____	62
Tableau 4.3. Niveaux hiérarchiques de la simulation analogique _____	65
Tableau 6.1. Limites de fonctions objectives pour un convertisseur électrothermique fabriqué en technologie AMS $0,6 \mu\text{m}$. _____	106
Tableau 6.2. Minima et maxima géométriques des différents paramètres de conception. _____	107
Tableau 6.3. Paramètres de conception initialisés aux valeurs du point de départ pour l'algorithme de la plus forte pente descendante. _____	112
Tableau 6.4. Paramètres de conception initialisés aux valeurs du point de départ pour l'algorithme de la plus forte pente descendante. _____	116
Tableau 6.5. Paramètres de conception initialisés aux valeurs du premier point de départ pour l'algorithme de la plus forte pente descendante. _____	123
Tableau 6.6. Paramètres de conception initialisés aux valeurs du deuxième point de départ pour l'algorithme de la plus forte pente descendante. _____	124

1 Introduction

On peut définir les microsystèmes ou les systèmes micro-électromécaniques comme des systèmes de petites dimensions (plus petites qu'un centimètre cube) et qui accomplissent des fonctions de précision. Les microsystèmes se composent de capteurs pour acquérir les informations du monde extérieur, d'une partie électronique pour le traitement de données et d'actionneurs qui réagissent avec le monde extérieur.

Les capteurs et les actionneurs sont fabriqués sur des microstructures de silicium, tandis que les parties de traitement de signal sont fabriquées avec les processus microélectroniques ordinaires.

Le terme microsystème est utilisé pour décrire des systèmes hybrides dont les différents composants sont fabriqués indépendamment (avec différentes technologies et différents substrats), ainsi que pour décrire des microsystèmes monolithiques dont les composants sont fabriqués sur le même substrat. L'avantage de ces systèmes hybrides réside dans leurs bonnes fonctionnalités assurées par des coûts de fabrication relativement faibles. La Figure 1.1 montre une représentation schématique d'un microsystème composé d'un capteur / actionneur et de l'électronique nécessaire.

Les technologies de fabrication de microsystèmes sont dérivées de celles de fabrication de la microélectronique. Ces technologies utilisent les mêmes techniques comme la génération de masque, la photolithographie, le micro-usinage en volume et le micro-usinage en surface, etc. Leur coût de fabrication relativement faible est assuré par des techniques de « *post-processing* » où des milliers de composants peuvent être fabriqués sur le même wafer de silicium.

Par contre, les coûts de prototypage de microsystèmes sont élevés. Les nombreuses itérations nécessaires pour la production d'un seul composant microsystème qui varient entre la modélisation, la simulation et l'optimisation de composant microsystème, rendent cette opération très coûteuse. D'où vient la nécessité d'une nouvelle génération d'outils CAO¹ pour microsystème capable de réduire le temps et l'effort de conception.

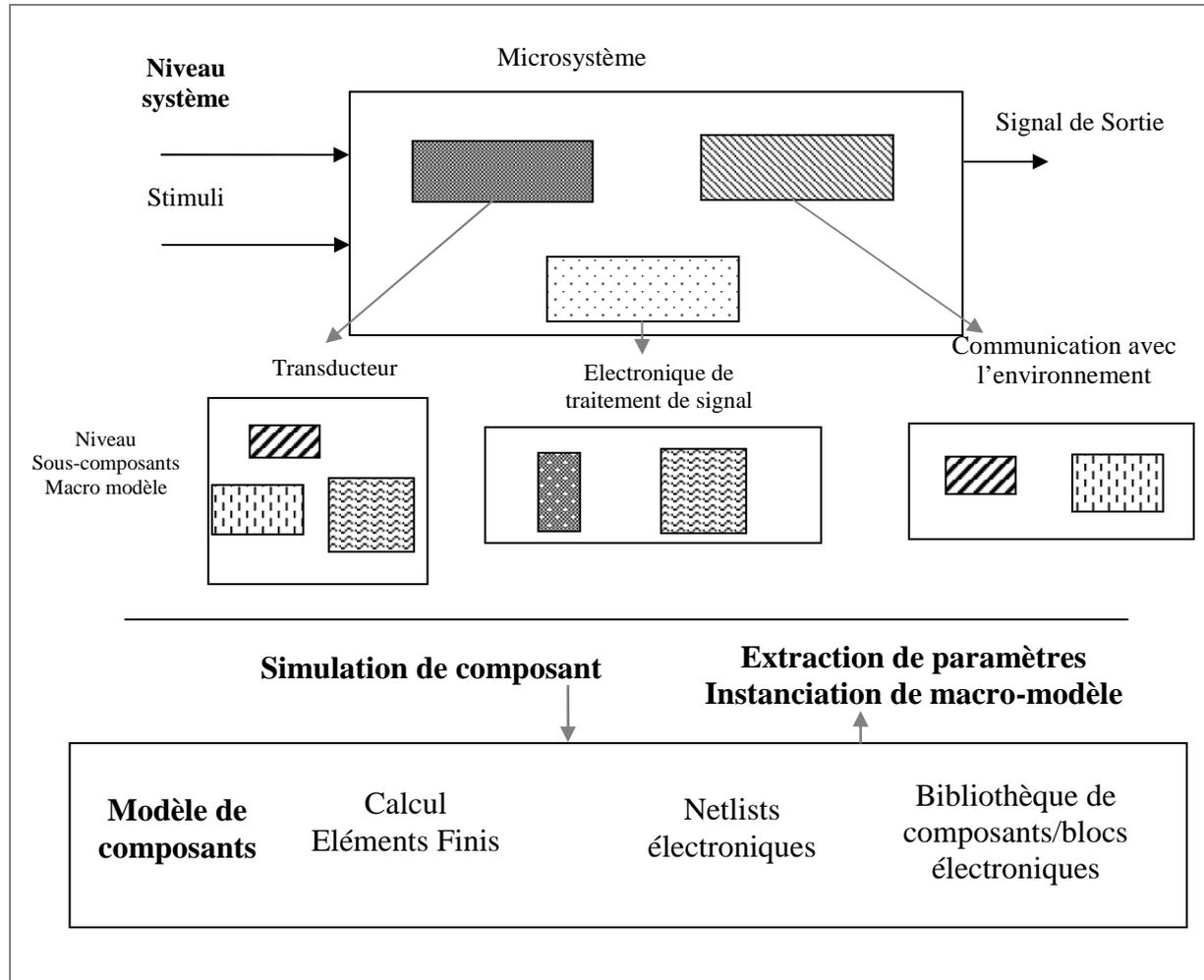


Figure 1.1. Représentation schématique d'un microsystème qui montre l'interaction entre les différents composants (capteurs, actionneurs et l'électronique nécessaire) et l'environnement. Différents niveaux de modélisation et simulation sont aussi montrés.

Comme dans le cas des technologies de fabrication, les outils CAO de microsystèmes sont basés sur des outils CAO de la microélectronique. Pour les adapter aux besoins des technologies des microsystèmes, les simulateurs de procédé technologique de la microélectronique sont étendus par des simulateurs de gravure capables de produire la géométrie 3D de composants. Le comportement dynamique de systèmes non-électrique est simulé par des simulateurs de circuits en construisant leurs modèles en circuits équivalents. Les outils existants de simulation et les modélisations dans les domaines mécaniques et thermiques peuvent être adaptés pour élaborer les composants de petites dimensions.

A chaque niveau de conception des microsystèmes, il existe des outils spécifiques pour la modélisation et la simulation des microsystèmes qui suivent le flot de conception :

¹ CAO Conception Assistée par Ordinateur

1. Le dessin du masque et la conception au niveau du layout : ce sont les éditeurs de layout et les outils de vérification de règles de dessin (DRC²).
2. La simulation de procédé technologique : les simulateurs de procédé semi-conducteur et les simulateurs de gravure.
3. La simulation de composant : les simulateurs Eléments Finis (FEM³, BEM⁴), circuits équivalents et les langages de description du matériel (HDLs⁵)
4. La simulation au niveau système : circuits équivalents, HDLs, simulateurs analogiques et mixtes analogiques/numériques.
5. Vérification et mesures : validation de la simulation.
6. Simulation globale : l'exécution itérative des étapes précédentes (Optimisation).

Il y a beaucoup d'outils de simulation qui existent sur ces différents niveaux de simulation [1, 3, 4], mais pour avoir un environnement complet pour la simulation et la modélisation de microsystèmes, il y a des exigences qui doivent être achevées :

- Un couplage entre les simulateurs de gravure 3D, les simulateurs de composant, les éditeurs de masque et les simulateurs de circuits.
- La possibilité des transferts souples des données entre les différents niveaux de simulation.

Les principaux défis que l'on peut affronter pour réaliser les tâches précédentes peuvent être résumés par [5] :

- La construction de modèles 3D de composant microsysteme à partir de son layout.
- La modélisation FEM de composants microsysteme.
- La construction de modèles à constantes localisées.
- L'insertion de ces macro-modèles dans un environnement dynamique de simulation.

L'étude du comportement dynamique de microsystèmes est complexe. Cette complexité vient du fait que la plupart des transducteurs ont des comportements non linéaires, ils font intervenir plusieurs domaines d'énergie et ils effectuent des actions complexes qui ajoutent une non linéarité géométrique au problème.

Dû aux coûts élevés de prototypage, la simulation joue un rôle important dans le domaine des microsystèmes et de la microélectronique. La modélisation analytique et les techniques de FEM sont utilisées dans le domaine de simulation de microstructures [6].

Le développement des outils CAO spécialement dédiés aux microsystèmes a commencé au début des années 80. Les premiers outils ont été essentiellement conçus pour améliorer la conception de layout des microsystèmes [7] [8] [9] [10]. MEMCADTM, SENSIM, CAPSIM, CAEMEMS-D, SENSOR, NM/SESESTM sont des exemples de ces outils. La plupart de ces outils utilisent des bases de données simples qui contiennent les propriétés des matériaux et les critères de performance. La communication entre les différents modules de ces outils se fait par des simples transferts de fichiers [11]. Des outils de simulation de composant spécifique ont été développés, l'outil PUSI [12] est un exemple de ces outils qui optimisent le comportement dynamique d'une micro pompe.

² DRC *Design Rules Checker*

³ FEM *Finite Element Method*

⁴ BEM *Boundary Element Method*

⁵ HDL *Hardware Description Language*

Les techniques de simulation électrique (de type SPICE) sont utilisées pour les circuits analogiques. Les techniques de simulation événementielles « *event-driven* » (exemple VHDL) sont utilisées pour les circuits numériques.

Le terme (simulation multi-niveaux) « *muti-level* » est employé pour décrire des simulations de différents niveaux d'abstraction. Le terme (simulation de nature mixte) « *mixed-nature* » est utilisé quand des signaux de natures différentes (électrique, mécanique, thermique, etc.) sont impliqués. Finalement le terme « *mixed-mode* » est utilisé pour décrire les simulations de modèles analogiques et numériques des circuits électroniques.

Les simulateurs SPICE peuvent être utilisés comme des « *solvers* » analogiques de modèles à constantes localisés « *lumped parameters model* » pour simuler les transducteurs en exploitant les analogies (électrique-mécanique, électrique-thermique, etc.) ce qui est l'approche des circuits équivalents. Quand ils sont instantiés dans une « *netlist*⁶ » avec l'électronique, ces modèles peuvent être utilisés pour prédire la performance globale du microsysteme. Les équations différentielles d'un composant physique sont remplacées par celles d'un circuit électrique équivalent (qui présente les mêmes équations). Un problème difficile s'impose quand on modélise un composant physique extrêmement non linéaire avec des points de discontinuité. La technique courante pour résoudre ce problème consiste à linéariser les composants autour d'un point de fonctionnement « *bias* » [13]. L'inconvénient de cette technique est qu'elle limite l'analyse de performance sur des petites intervalles de signal.

1.1 Etat de l'art

On présente dans cette section les différentes approches et les différents outils qui font le couplage entre la simulation au niveau composant et la simulation au niveau système.

CoSolve-EMTM, un "*solver*" qui couple des simulations 3D quasi-statique pour les microsystemes électromécaniques est présenté dans [14]. Cinq classes de problèmes électromécaniques sont traitées :

1. Composants avec instabilité électrostatique de type "*pull-in*".
2. Composants dont des déformations précises sont exigées.
3. Composants actionnés par des conducteurs multiples.
4. Capteurs capacitifs qui utilisent des contacts de surface.
5. Actionneurs qui utilisent des contacts de surface.

Cet outil est intégré dans MEMCADTM platform qui est distribué par *Coventor Technology Inc.*

L'outil SENSOR [8] fournit un générateur automatique de macro modèle d'un capteur mécanique ou thermique qui est simulé dans l'étape suivante par H-SPICETM ou par ELDOTM. Les paramètres à constantes localisées du modèle sont calculés soit par des méthodes analytiques, soit par des simulations Eléments Finis (FEM). Des simulations simplifiées des caractéristiques de sortie (y compris des comportements non linéaires) sont possibles.

La modélisation et la simulation de transducteurs microsystemes en utilisant des circuits équivalents à constantes réparties sont illustrées dans [13]. Des transducteurs micro-

⁶ *Netlist* : Fichier de description d'un circuit électrique organisé en nœuds et branches

électro-mécaniques sont décrits en prenant en compte leurs comportements dynamiques. La limitation de cette approche réside toujours dans l'utilisation de modèles à paramètres constants qui sont linéarisés autour d'un point de fonctionnement. La méthodologie de modélisation est étendue pour décrire deux modèles à constantes réparties.

Les simulateurs SPICE peuvent être utilisés pour simuler des systèmes à constantes réparties, cette approche est discutée dans [13]. Un modèle à constantes réparties est présenté dans [15], il s'agit de la modélisation de l'amortissement dans les gaz dans une microstructure. Dans cette approche il s'agit de définir un maillage discret de la microstructure avec les conditions aux limites appropriées "*boundary conditions*". Les éléments de maillage sont les composants électriques élémentaires. Cette méthode peut être envisagée pour modéliser des systèmes extrêmement non linéaires avec une grande précision. On peut reprocher à cette méthode d'être compliquée et difficile à implémenter tant qu'un outil de maillage automatique n'est pas accessible.

Les outils CAO microélectroniques commerciaux disponibles [16] ont besoin d'être modifiés avant de les utiliser pour la conception de microsystèmes.

Le travail de l'équipe microsystème du laboratoire TIMA-CMP s'est focalisé sur l'extension de l'environnement de CAO pour qu'il soit capable de manipuler un flot de conception microsystème [17] [18]. Dans cet environnement, plusieurs facilités sont données au concepteur microsystème :

- Génération de layout d'un microsystème qui contient des parties électroniques et des parties non électroniques à partir des bibliothèques en format CIF⁷ ou GDSII⁸.
- Vérification des règles de dessin de microsystème et de microélectronique.
- Extraction des paramètres de niveau layout vers le niveau *netlist*.
- Possibilité de reconnaître les composants microsystème (poutre, membrane, cantilever) par l'extracteur.
- Possibilité de simuler la *netlist* générée en utilisant des modèles HDL-A paramétrables.
- Bibliothèque de cellules paramétrables de composants microsystème.

L'approche est valable pour les microsystèmes monolithiques où la connaissance de la technologie CMOS est intense. Par contre l'application d'une telle approche sur des microsystèmes hybrides est problématique.

Le couplage entre les *solvers FE* et les simulateurs de circuits est décrit dans [19]. Deux simulateurs (ANSYS[®] et PSPICE) sont couplés pour calculer la réponse d'une poutre vibrante. Le couplage entre les deux simulateurs n'est pas intrinsèque, mais il se fait itérativement en mettant à jour la réponse de chaque simulateur. Cette approche a l'inconvénient d'être valable uniquement sur des cas simples. Les problèmes de convergence ne sont pas discutés pour des cas plus compliqués. De plus, un tel couplage est très coûteux en terme de temps et de calcul.

Une étude sur l'optimisation des microsystèmes intégrés basée sur le couplage du simulateur au niveau composant et l'analyse de stabilité de système est présentée dans [20]. Les microstructures étudiées sont deux convertisseurs thermiques composés d'une membrane conçue avec une seule couche d'oxyde.

⁷ CIF Caltech Intermediate Format data exchange format

⁸ GDSII Graphical Drawing Standard

La génération d'un modèle comportemental à partir des simulations FE est discutée dans [21]. Des analyses transitoires à partir des stimuli standards sont appliquées, la sortie qui dépend du temps est ensuite analysée. Si le comportement analysé coïncide avec le comportement exigé, un modèle HDL est généré. Dans le cas contraire, on applique des algorithmes récursifs d'optimisation. La première difficulté en appliquant cette approche est le besoin d'avoir une bibliothèque de modèles FE avant d'effectuer les analyses transitoires. La deuxième difficulté est la complexité d'interpréter le comportement transitoire, et finalement les analyses FE modales et transitoires sont des calculs très lourds et très coûteux au niveau du temps de calcul.

Le couplage des simulateurs FE et les simulateurs de circuits est un domaine très populaire. Malheureusement, cette approche reste limitée sur de simples composants dont le nombre de degrés de liberté (DOFs⁹) est limité. Plusieurs exemples sur le couplage des simulateurs FE et des simulateurs de circuits sont démontrés dans [22] [23] [24] [25].

1.2 Présentation de la thèse

Cette thèse concerne les outils CAO pour microsysteme. Les différentes approches pour automatiser la conception de microsysteme en prenant en compte les deux flots possibles (*bottom-up* et *top-down*) sont étudiées dans un aperçu global de l'état de l'art. C'est le premier chapitre introductif.

Afin de se repérer par rapport au flot de conception microélectronique, une comparaison entre le flot conception microsysteme et le flot de conception microélectronique est discutée dans le second chapitre.

Le troisième chapitre donnera un aperçu des différentes technologies de fabrication des microsystemes en se focalisant plus spécialement sur les technologies intégrées avec l'électronique. L'intérêt de ce chapitre est de montrer les effets du procédé technologique modélisé et simulé sur les outils CAO utilisés pour la simulation et la modélisation.

Le chapitre 4, consacré lui directement à la modélisation et simulation de microsystemes, tentera de définir une méthodologie de modélisation et de l'appliquer à quelques exemples. Ce chapitre se focalisera surtout sur la problématique de la modélisation des microsystemes et tentera de trouver quel niveau de modélisation est le plus adapté à la synthèse de microsystemes pour permettre d'adapter des outils spécifiques de simulation déjà existants. Un environnement de simulation globale pour microsysteme est détaillé dans à la fin de ce chapitre tentera de définir une méthodologie permettant de coupler plusieurs simulateurs afin de simuler un composant microsysteme avec son environnement (SoC¹⁰).

Le cinquième chapitre est consacré au dessin des masques et aux règles de dessin pour microsystemes. Dans ce chapitre nous exposons les différents problèmes entraînés par l'usage intensif des formes non-manhattan dans le domaine de la conception des microsystemes. Nous proposons des méthodologies pour générer des layout compatibles avec les éditeurs de layout. Nous proposons aussi des algorithmes de vérification de règles de dessin bien adaptés aux formes non-manhattan en particulier et aux règles de dessin pour microsystemes en général.

⁹ DOFs *Degrees Of Freedom*

¹⁰ SoC *System On Chip*

Dans le chapitre 6, nous introduisons une méthodologie de synthèse et d'optimisation pour microsystemes. Un environnement de description comportementale et géométrique est développé à travers ce chapitre. La méthodologie est validée sur un composant microsysteme qui est le convertisseur électrothermique. La deuxième partie de ce chapitre est consacrée aux méthodes d'optimisation de microsystemes. Différentes méthodes d'optimisation sont discutées puis appliquées sur le convertisseur électrothermique.

Un chapitre de conclusion générale viendra clore ce manuscrit.

Références de l'introduction

- [1] Stephen D. Senturia, "CAD challenges for microsensors, microactuators and microsystems", *Proceeding of IEEE*, vol. 86, pp. 1611-1626, 1998.
- [2] Lynn D. Gabbay and Stephen D. Senturia, "Automatic generation of dynamic macro-models using quasistatic simulation with modal analysis", in *Tech. Dig. IEEE Solid – State Sensor and Actuator Workshop*, pp.197-200, Hilton Head, SC, USA, June 1988.
- [3] Stephen D. Senturia, "Simulation and design of microsystems: a 10 years perspective", *Sensors and actuators*, vol. A67, pp. 1-7, 1998.
- [4] Gerhard Wachutka, Herbert Pavlicek, Thomas Fabula, Henning Haffner, Nicolaus Hey, Thomas Feudel, and Robert Strecker, *CAD tools for MEMS*. UETP-MEMS Course. FSRM, Rue de l'Orangerie 8, 2000 Neuchâtel, Switzerland, 1994.
- [5] Stephen D. Senturia, "CAD for microelectromechanical systems", in *8th International Conference of Solid-State Sensors and Actuators, and Eurosensors IX. Digest of Technical Papers*, vol. 2 (of 2), pp. 5-8, Stockholm, Sweden, 1995.
- [6] B. Puers, E. Peeters, and E. Sansen, "CAD tools in mechanical sensor design", *Sensors and Actuators*, vol. 17, pp. 423-429, 1989.
- [7] H. U. Schwarzenbach, J. G. Korvink, M. Roos, G. Sartoris, and E. Anderheggen, "A micro electro mechanical CAD extension to SESES", *Journal of Micromechanics and Microengineering*, vol. 3, pp. 191-200, 1993.
- [8] B. Folkmer, H. L. Offereins, H. Sandmaier, W. Lang, A. Seidl, P. Groth, and R. Pressmar, "A simulation tool for mechanical sensor design (SENSOR)", *Sensors and Actuators*, vol. A32, pp. 521-524, 1992.
- [9] K. W. Lee and K. D. Wise, "SENSIM: A simulation program for solid state pressure sensors", *IEEE Transactions on Electron Devices*, vol. ED-29, pp. 34-41, 1992.
- [10] S. D. Senturia, R. Harris, S. Kim, K. Nabors, M. Shulmann, and J. White, "A computer aided design system for microelectromechanical systems (MEMCAD)", *Journal of Microelectromechanical systems*, vol. 1, pp. 3-13, 1992.
- [11] B. Folkmer and H. Sandmaier, *Simulation tools for micro electro mechanical systems, Simulation and Design for Microsystems and Microstructures*. Computational Mechanics Publications, Ashurst Lodge, Ashurst, Southampton, SO40 7AA, United Kingdom, 1995.
- [12] R. Zengerle, M. Richter, F. Brosinger, A. Richter, and H. Sandmaier, "Performance simulation of microminiaturized membrane pumps", In *7th International Conference on Solid-State Sensors and Actuators*, pp. 106-109, Yokohama, Japan, 1993.

- [13] A. Harrie, C. Tilman, "Equivalent circuit representation of electromechanical transducers: I. lumped-parameter systems (micromechanical)", *Journal of Micromechanics and Microengineering*, vol. 6, pp. 157-176, 1995.
- [14] J. R. Gilbert, R. Legtenberg, and S. D. Senturia, "3D coupled electromechanics for MEMS: applications of CoSolve-EM", *In Proceedings. IEEE Micro Electro Mechanical Systems*, pp. 122-127, Amsterdam, Netherlands, 1995.
- [15] T. Veijola, T. Ryhanen, H. kuisma, and J. Lahdenpera, "Circuits simulation model gas damping in microstructures with non trivial geometry", *in 8th International Conference on Solid-State Sensors and Actuators, and Eurosensors IX, Digest of Technical Papers*, vol. 2, pp. 36-39, Stockholm, Sweden, June 1995.
- [16] Cadence Design Systems, 555 River Oaks Parkway, San Jose, CA 95134, USA, *CADENCE, Marketing services*, 1997
- [17] J. M. Karam, B. Courtois, and M. Bauge, "High level CAD melds microsystems with foundries", *in Proceedings, European Design and Test Conference ED&TC 96*, pp. 442-447, Paris, France, March 1996.
- [18] J. M. Karam, B. Courtois, A. Poppe, K. Hofmann, M. Rencz, M. Glesner, and V. Szekely, "Applied design and analysis of microsystems", *in Proceedings, European Design and Test Conference*, pp.528-532, 1996.
- [19] A. Schroth, T. Blochwitz, and G. Gerlach, "Simulation of a complex sensor system using coupled simulation programs", *Sensors and Actuators A (Physical)*, vol.A54, pp. 632-635, 1996.
- [20] D. Jaeggi, J. Fink, A. Haberli, and H. Baltes, "Overall system analysis of a cmos thermal converter", *In 8th International Conference on Solid-State Sensors and Actuators, and Eurosensors IX, Digest of Technical Papers*, vol. 2, pp. 112-115, Stockholm, Sweden, June 1995.
- [21] K. Hofmann, M. Lang, J. M. Karam, M. Glesner, and B. Courtois, "Generation of a behavioral model of an acceleration sensor from its finite-element-description", *In 1st Europe-Asia Congress on Mechatronics*, pp. 64-69, Besançon, France, October 1996.
- [22] K. Richard, T. L. Sevilla, and M. C. Rodamaker, "Predicting hysteric oscillations in over-temperature protection of a power IC using transient electrothermal circuit simulation", *In Proceedings of the 5th International Symposium on Power Semiconductor Devices and ICs*, pp. 304-309, Monterey, CA, USA, November 1993.
- [23] J. Vaananen, "Circuit theoretical approach to couple two-dimensional finite element models with external circuit equations", *IEEE Transactions on Magnetics*, vol. 32, pp. 400-410, Mar 1996.
- [24] J. G. Rollins and J. J. Choma, "Mixed-mode PISCES-SPICE coupled circuit and device solver", *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, vol. 7, pp. 862-867, 1988.

- [25] J. Litsios, S Müller, and W. Fichtner "Mixed-mode Multi-dimensional devices and circuit simulation", Technical report no. 93/28, Integrated Systems Laboratory, Swiss Federal Institute of Technology Zurich, 1993.

2

Flot de conception microsysteme

Pour transformer les microsystemes, de prototypes de recherche en produits industriels disponibles sur le marche, des procedes de fabrication ainsi que des outils CAO doivent etre disponibles. Ces outils gèrent de façon efficace les grandes quantités de données qui sont nécessaires au processus de conception. Ils permettent notamment de modéliser les microcomposants à différents niveaux d'abstraction et de simuler leur comportement.

Pour les différentes disciplines d'ingénierie, telles que l'électronique, la mécanique, l'optique, et d'autres, il existe une variété d'outils de conception, de simulation et de vérification d'un composant spécifié : le manque, aujourd'hui est de définir un environnement dans lequel une conception complète de microsysteme peut etre achevée de façon systematique.

En general, pour définir un tel environnement, deux approches peuvent etre envisagées [11] [12] : la première consiste à développer un nouvel environnement spécifique aux microsystemes ; la seconde plus réaliste réside dans l'utilisation d'outils CAO existant pour les différentes disciplines requises par les microsystemes, de les étendre et de les interfacer

dans un seul environnement, avec des cellules de composants standards et paramétrables, afin d'assurer un flot de conception "*design flow*" systématique et continu.

Les outils de conception et de simulation microélectronique couvrent un besoin large qui peut satisfaire à des phases particulières de la conception d'un microsysteme. Cependant ces outils devront être étendus et ceci à des niveaux différents de manière à considérer plusieurs domaines d'énergie, en plus du domaine électrique. Ceci a des conséquences importantes dans la méthodologie de conception qui devra considérer d'autres outils en plus de ceux qui sont utilisés dans le cas purement microélectronique [11].

Dans ce chapitre, nous faisons l'état de l'art d'un flot de conception microsysteme, ce qui permettra de repérer l'emplacement tout au long de ce flot des différents outils comme d'identifier les données à l'entrée et à la sortie de chaque outil. Nous comparons le flot de conception microélectronique au flot microsysteme, qui sont évidemment liés l'un à l'autre. Par la suite, nous nous focaliserons sur les outils de conception des microsystemes et, éventuellement, de l'ensemble microsysteme et microélectronique quand les deux flots convergent. Nous commencerons par les outils nécessaires aux niveaux d'abstraction les plus bas et procéderons en amont.

2.1 Structure globale d'un environnement de conception microsysteme

L'environnement d'outils de CAO pour microsysteme permet un flux de conception continu qui peut être vu sous deux angles différents : celui du concepteur système n'ayant pas de connaissances pointues dans le domaine des microsystemes, et celui du concepteur de composants ayant l'expérience dans le domaine. De ce fait, l'environnement comporte des éléments utiles pour le concepteur de composants lui permettant de concevoir des modules, de les simuler et de finalement traduire ses résultats et son expérience, par des modèles standard caractérisés stockés dans la bibliothèque, à l'utilisation du concepteur de systèmes (voir la Figure 2.1).

Cet environnement couvrant les différents niveaux de modélisation et de simulation de microsystemes permet les fonctions suivantes [11] [14] :

- Un flux continu de conception moyennant l'extension (kit de conception) des outils de conception des circuits intégrés déjà disponibles sur le marché permettant la génération de dessin des masques (layout) d'un microsysteme comportant des parties électroniques et non-électroniques et la vérification des règles de dessins, et aussi l'extraction de paramètres du niveau layout au niveau plus haut afin de faire des simulations post-layout "*back annotation*",
- Une vérification globale de la fonctionnalité du système en utilisant un simulateur mixte, en ajoutant des éléments de nature non-électrique, des éléments d'interface et les stimuli correspondants.

Les méthodologies de conception des circuits intégrés reposent sur des approches très structurées. Elles sont basées sur la validation par simulation à différents niveaux hiérarchiques de représentation et utilisent des bibliothèques de composants dans un environnement CAO. Typiquement, la méthodologie de conception est "*top-down*", depuis les niveaux d'abstraction plus hauts, avec moins de détails, vers les niveaux d'abstraction plus bas où tout le détail du circuit ou du dispositif est disponible. A l'inverse, la méthodologie de conception "*bottom-up*" commence par les niveaux d'abstraction les plus bas et procède par

abstraction vers les niveaux plus hauts. Ces approches très structurées servent à réduire au maximum le nombre de cycles de conception-fabrication-test et par voie de conséquence le temps de conception.

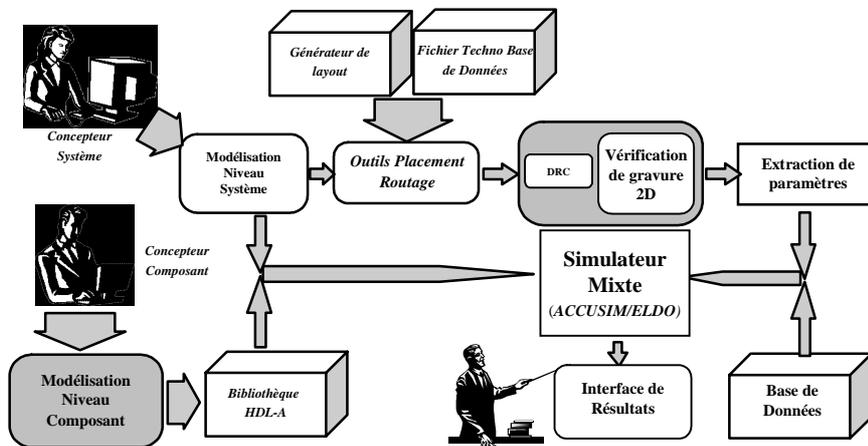


Figure 2.1. Structure globale d'un environnement de conception microsystème

2.2 Flot de conception microélectronique

La Figure 2.2 montre le flot de conception microélectronique. Au plus bas niveau, nous avons le dessin des masques de lithographie¹¹. A partir de ces masques, les outils que l'on appelle de T-CAD¹² permettent de simuler le procédé de fabrication et la physique des microcomposants [1]. Dans la microélectronique, il existe un nombre réduit de composants (transistors, condensateurs, résistances, etc.) et donc de modèles, ces composants étant utilisés en grand nombre dans un circuit. Procédant vers le haut, nous trouvons ensuite le niveau circuit. Les composants sont ici interconnectés en formant un réseau électrique, chaque composant étant décrit par un modèle de son comportement. Des outils appelés outils d'extraction du layout permettent d'identifier les composants et d'en tirer automatiquement une *netlist* décrivant le circuit. Inversement, les outils de synthèse physique, qui comportent des outils de placement de cellule et de routage des connexions, permettent de générer le layout à partir d'une *netlist* [2]. A ce niveau, la simulation du circuit est faite au moyen des simulateurs analogiques tels que SPICE. Par la suite, nous trouvons des niveaux de représentation plus abstraits, numériques ou analogiques où la description du comportement se fait principalement au moyen de langages de description du matériel HDL qui sont aujourd'hui devenus des standards [4][16].

¹¹ Layout

¹² TCAD Technology Computer-Aided Design

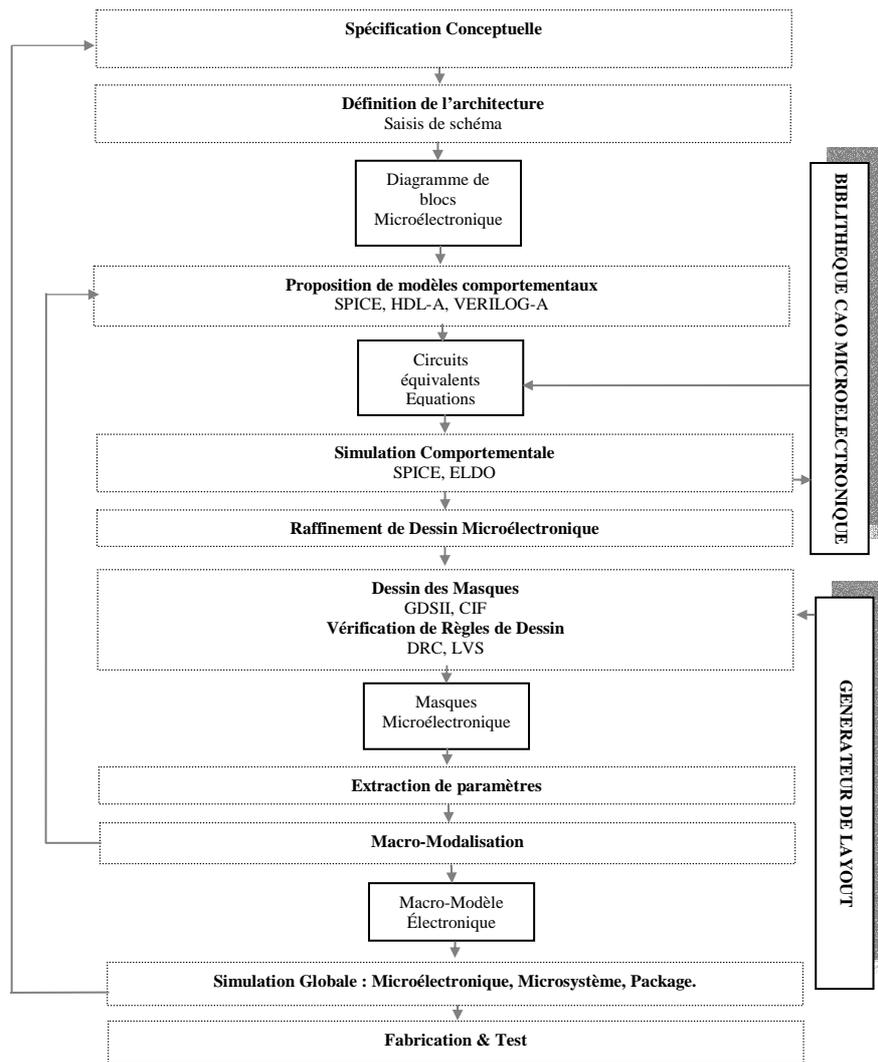


Figure 2.2. Flot de conception Microélectronique.

Si les méthodologies de conception sont très bien structurées dans le cas des circuits numériques, elles le sont déjà moins dans les cas des circuits analogiques et mixtes. L'intégration des microsystemes, avec l'incorporation de multiples domaines d'énergie et des nouvelles étapes dans le procédé de fabrication, va introduire des changements importants dans le flot de conception [6] [8].

2.3 Flot de conception microsysteme

La Figure 2.3 montre comment ce flot peut être organisé depuis la spécification conceptuelle du microsysteme en haut vers la fabrication et le test du dispositif en bas. Ce flot servira de fil conducteur à notre propos afin de repérer l'emplacement des différents outils de la CAO, en fonction de l'avancement dans la conception et les différents niveaux d'abstraction

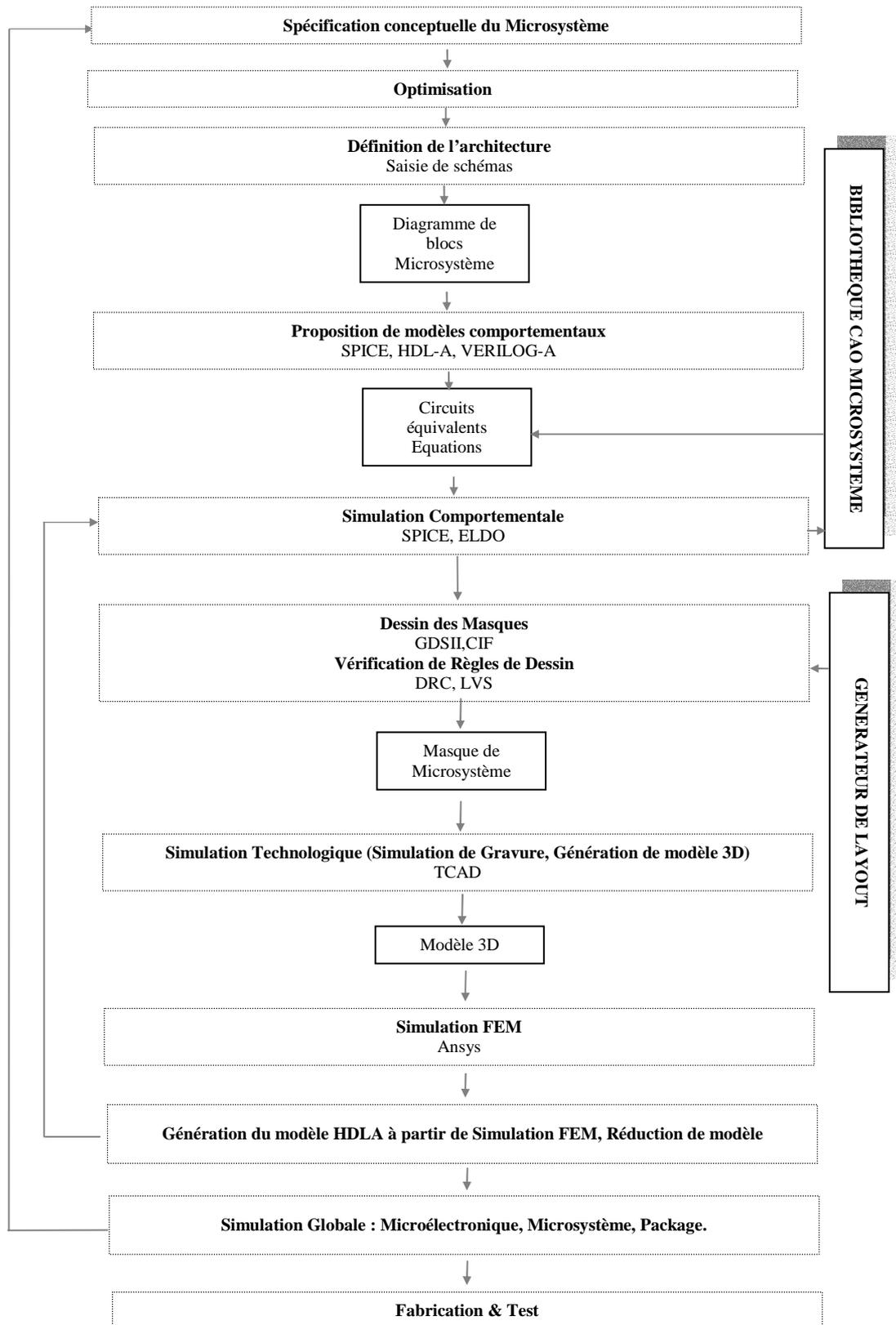


Figure 2.3. Flot de conception des microsystemes.

La modélisation et l'analyse des composants et systèmes est une tâche délicate. Il existe plusieurs niveaux de modélisation et les concepts fondamentaux varient considérablement d'un niveau à l'autre [10] [13].

Sur la Figure 2.3 nous proposons un schéma de ces niveaux. On y distingue quatre niveaux qui sont : le niveau système, le niveau composant, le niveau physique et le niveau process. Les doubles flèches de liaison indiquent un échange itératif d'informations inter-niveaux.

Le niveau le plus élevé est celui du système. C'est le champ d'utilisation des techniques de description par les blocs comportementaux (*behavioral*) et des éléments de circuit localisé. Ces deux techniques conduisent à un ensemble d'équations différentielles ordinaires pour décrire le comportement dynamique du système. Souvent le système d'équations différentielles est écrit sous la forme d'un système couplé du premier ordre en fonction d'un ensemble de variables appelées variables d'état. Sous cette forme le système d'équations est appelé équations d'état [15].

Tout en bas de la Figure se trouve le niveau process. C'est ici que les étapes du procédé de fabrication ainsi que les masques sont créés. La modélisation du procédé est une activité très sophistiquée pour laquelle plusieurs outils de CAO ont été développés sous le nom générique de technologie CAD ou TCAD. Pour le concepteur de microsysteme, l'importance d'un TCAD provient essentiellement de sa capacité à prédire la géométrie d'un composant à partir du jeu de masques et des séquences du procédé. Aussi, le concepteur doit avoir une idée suffisamment fine du procédé pour pouvoir adapter les propriétés des matériaux utilisées à chaque variante du process.

Le niveau physique de modélisation adresse le comportement du composant réel dans un continuum à 3 dimensions. Les équations régissant ce domaine sont les équations aux dérivées partielles. Différentes méthodes analytiques peuvent être utilisées pour trouver des solutions explicites dans les cas des géométries idéales. Cependant, la modélisation des composants réels fait très fréquemment appel à des solutions numériques basées sur un maillage dense du domaine physique. Il existe plusieurs outils commerciaux s'appuyant soit sur les méthodes d'éléments finis, soit sur les méthodes de différences finies, soit celles des éléments frontières.

Malgré l'intérêt que représentent ces représentations maillées des équations aux dérivées partielles pour la modélisation physique, leur usage est prohibitif quand on souhaite traiter les composants et l'électronique qui leur est associée. C'est ici que la modélisation au niveau composant intervient par la création de macro modèles (*Macro Models*) ou de modèles d'ordre réduit (*Reduced Order Models*) dans une forme qui capture l'essentiel du comportement physique du composant d'un système et qui est simultanément compatible avec la description au niveau système.

Un macro modèle idéal est plutôt analytique que numérique. Il doit en plus capturer les caractéristiques essentielles du composant dans une forme compatible avec les simulateurs de type système. Le macro modèle doit aussi satisfaire des exigences énergétiques en étant conservateur d'énergie ou dissipateur en accord avec ce qu'est le composant physique représenté. Il doit aussi reproduire la dépendance du composant physique à la géométrie et les propriétés des matériaux.

2.4 Le flot de conception microsysteme face au flot de conception microelectronique

Nous devons d'abord remarquer que la conception des microsystemes n'est pas aussi bien structurée que dans le cas de la microelectronique. En effet, la diversité de microsystemes et de phénomènes physiques à prendre en compte étant trop grande, nous n'aurons pas en général de bibliothèques de composants détaillant de manière assez précise leur fonctionnement. Par conséquent, la simulation au niveau physique des microcomposants fait partie intégrante du flot de conception [9] [11] [13]. Nous pouvons alors espérer un certain nombre d'itérations entre les différentes étapes du flot de conception et, très certainement, plusieurs cycles de conception-fabrication-test.

Notons aussi que le flot de conception microelectronique et le flot de conception microsysteme sont reliés dans les étapes qui demandent la validation de l'ensemble microsysteme. Dans la plupart des étapes, le concepteur peut utiliser des données ou des modèles disponibles dans les bibliothèques de l'environnement de CAO. Notons aussi que le flot de conception est initialement *top-down*, depuis la spécification vers les dessins de masque. Etant donné que les masques sont obtenus séparément dans le flot microelectronique et le flot microsysteme, un flot *bottom-up* depuis les masques vers une validation globale du microsysteme est nécessaire.

Le flot de conception commence par une spécification conceptuelle du microsysteme, en particulier son comportement vis-à-vis de l'environnement. Le packaging joue alors un rôle primordial dans le dispositif final, car il devra permettre aux microsystemes de percevoir et d'agir sur l'environnement. Ayant aussi un fort impact sur le comportement global et sur le coût total, il devient indispensable dès lors de considérer cet aspect tout au début de la conception.

Par la suite, l'architecture du microsysteme peut être définie à l'aide de diagrammes de blocs, autant pour la partie microelectronique que pour la partie microsysteme. Cette tâche dans un environnement de CAO est couramment appelée *saisie de schémas*. La validation de l'architecture demande des modèles comportementaux. Pour la partie microelectronique ces modèles sont normalement accessibles dans les bibliothèques, sinon il faudra développer une description avec un langage HDL.

Dans certains cas, la partie microsysteme pourra aussi utiliser des modèles existant dans les bibliothèques. Mais très souvent nous serons amenés à développer ces modèles à partir d'une analyse mathématique, aboutissant soit à des circuits équivalents soit à un ensemble d'équations.

La simulation comportementale de l'ensemble du microsysteme est alors faite au moyen des simulateurs de circuit de type SPICE ou des simulateurs HDL.

Une fois que l'architecture du microsysteme est validée, une étape de conception en vue du test est impérative. Les microsystemes ayant des hauts niveaux d'intégration, des stratégies de test sont nécessaires afin de s'assurer que les dispositifs fabriqués fonctionnent correctement et sans demander des coûts de test excessifs.

Après la validation de l'architecture, la partie microelectronique sera progressivement détaillée au travers des différents niveaux d'abstraction que l'on a indiqué ci-dessus, afin d'obtenir les masques de fabrication. Des outils de synthèse sont, la aussi, largement

disponibles. Par contre, pour la partie microsysteme, il n'existe pas de methode de conception (ou de synthese) plus structuree. La conception des masques devient un procede fort creatif en fonction du comportement desiree et de la technologie de fabrication visee. Ce procede n'aboutira normalement qu'a une premiere proposition qui pourra etre raffinee une fois les simulations de plus bas niveau effectuees.

Les masques de fabrication permettent la simulation des procedes de fabrication avec les outils de TCAD. Ceci n'est generalement pas necessaire dans le cas de la microelectronique, car les modeles des composants (transistors) disponibles dans les bibliothèques sont très précis et ont été validés par des mesures électriques. Par contre cet aspect devient necessaire pour certains microsystemes étant donnée la grande diversite de composants.

La forme geometrique precise des microsystemes est parfaitement determinee à partir des masques de fabrication et la simulation du procede technologique. Par la suite, des calculs numeriques bases sur des modeles en elements finis FEM permettront la simulation des differents phenomenes physiques, en particulier les effets de champ dans les differents domaines d'energie intervenants. Cette simulation est la plus detaillee et la plus proche du dispositif reel. Il est aussi necessaire d'inclure dans cette simulation les charges ou contraintes imposees par le packaging.

L'existence de plusieurs domaines d'energie dans les microsystemes entraine l'utilisation de methodes de simulations couplees. Notons qu'il existe d'autres techniques de calcul numerique comme l'analyse en elements finis de frontiere BEM. Cette methode est analogue aux elements finis à ceci pres que seuls les contours du domaine sont maillés.

La simulation de l'ensemble du microsysteme au plus bas niveau de detail demande des outils de la CAO differents [13], comme les analyses FEM combines aux simulateurs electriques. Ces taches de co-simulation sont très lourdes à mettre en place, demandant des temps longs de simulation souvent inadmissibles. Des macro-modeles sont necessaires afin de pouvoir simuler ces systemes largement heterogenes.

Plus generalement, la necessite de simuler un ensemble de composants très heterogenes suscite à l'heure actuelle beaucoup d'interets de recherche. Ceci est evident dans un microsysteme, incorporant dans la même puce du materiel et du logiciel, ce premier pouvant etre du materiel electronique (numerique, analogique et mixte), mais aussi optique, thermique, mecanique, etc.

La fabrication finale du dispositif doit etre precedee d'une etape de simulation de fautes et de generation de vecteurs de test qui seront utilises lors du test de production et, parfois, lors de l'application. Les techniques de conception en vue du test inserees dans le microsysteme faciliteront le deroulement de cette etape qui est devenu imperative dans les systemes hautement integres.

Références du chapitre 2

- [1] M. Nagata, "Limitations, Innovations, Challenges of Circuits and Devices into a Half Micrometer and Beyond", *IEEE Journal of Solid-State Circuits*, vol. SC-27, April 1992, pp. 465-472.
- [2] K. R. Laker, W. M. C. Sansen, "Design of Analog Integrated Circuits and Systems". Mac Graw Hill, 1994.
- [3] H. P. Amann, P. Moeschler, F. Pellandini, A. Vachoux, C. Munk, and D. Mlynek,, "High-level specification of behavioral hardware models with MODES", *Proceeding of IEEE International Symposium on Circuits and Systems* , pp. 387-390, London, UK, 1994.
- [4] K. Bult, "Analog CMOS square-law circuits" Ph.D. dissertation, University of Twente, Enschede, The Netherlands, 1988.
- [5] J. Rijmenants, J. B. Litisios, T. R. Schwarz and M. G. R. Degrauwe, "ILAC: Automated Layout tool for Analog CMOS Circuits". *IEEE Journal of Solid-State Circuits*, vol. 24, n° 24, pp. 417-425, April 1989.
- [6] M. Kayal, S. Piguet, M. Declercq and B. Hochet, "SALIM: A Layout Generation Tool for Analog ICs", *IEEE Journal of Solid-State Circuits*, pp. 751-754, May 1988.
- [7] L.O. Donzelle, P. F. Dubois, B. Hennion, J. Parissis, P. Senn, "A constraint based approach to automatic design of analog cell", *In Proc. 28th DAC*, San Francisco, pp.506-509, June 1991.
- [8] J. M. Cohn, D. J. Garrod, R. A. Rutenbar, L. R. Carley, "Analog device-level Layout Automation", New York: Kluwer Academic, 1994.
- [9] M. Kayal, S. Piguet, M. Declercq and B. Hochet: SALIM : A Layout Generation Tool For CMOS Analog ICs", *Proc. CICC*, Rochester, May 1988, pp. 7.5.1-7.5.4
- [10] Stephen D. Senturia, "CAD challenges for microsensors, microactuators and microsystems", *Proceeding of IEEE*, vol. 86, pp. 1611-1626, 1998.
- [11] J. M. Karam, B. Courtois, and M. Bauge, "High level CAD melds microsystems with foundries", *in Proceedings, European Design and Test Conference ED&TC 96*, pp. 442-447, Paris, France, March 1996.
- [12] Stephen D. Senturia, "Simulation and design of microsystems: a 10 years perspective", *Sensors and actuators*, vol. A67, pp. 1-7, 1998.
- [13] Stephen D. Senturia, "CAD for microelectromechanical systems", *in 8th International Conference of Solid-State Sensors and Actuators, and Eurosensors IX. Digest of Technical Papers*, vol. 2 (of 2), pp. 5-8, Stockholm, Sweden, 1995.

- [14] B. Folkmer and H. Sandmaier, *Simulation tools for micro electro mechanical systems*, Simulation and Design for Microsystems and Microstructures. Computational Mechanics Publications, Ashurst Lodge, Ashurst, Southampton, SO40 7AA, United Kingdom, 1995.
- [15] A. Schroth, T. Blochwitz, and G. Gerlach, "Simulation of a complex sensor system using coupled simulation programs", *Sensors and Actuators A (Physical)*, vol.A54, pp. 632-635, 1996.

3 Technologies de fabrication et outils de simulation de procédé

Ce chapitre donne un aperçu détaillé des différentes technologies de fabrication des microsystemes pour introduire l'étude des outils CAO dédiés aux technologies microsysteme. Les technologies microsysteme ne sont pas aussi standardisées que les technologies de circuits intégrés et les développements technologiques sont plus souvent tirés par les applications. De ce fait les technologies microsystemes sont très différentes les unes des autres. Cependant il apparaît quand même des classes de technologies que l'on peut apparenter aux diverses technologies de circuits intégrés et qui supportent des applications différentes. C'est pourquoi nous allons nous intéresser aux deux technologies les plus courantes à savoir le micro-usinage en volume et le micro-usinage en surface. En ce qui concerne les applications, nous montrerons différents types d'applications classées par domaines d'applications comme le biomédical, l'automobile, le spatial, etc.

3.1 Les technologies microsystemes

3.1.1 Le micro-usinage en surface

Le micro-usinage en surface est basé sur l'emploi de couches sacrificielles. Ces couches vont être gravées entièrement à la fin du process par une étape de gravure sélective et isotrope. Le terme micro-usinage en surface vient par opposition à l'expression micro-usinage en volume car ici la gravure n'attaque pas le silicium du substrat (*bulk*). Pendant ce procédé les couches déposées vont être une succession de couches sacrificielles de dioxyde de silicium et de couches structurelles de polysilicium. Au cours du procédé les couches d'oxydes sacrificielles sont gravées de manière à définir les zones d'ancrage des structures en polysilicium sur le substrat ou sur le niveau inférieur. La Figure 3.1 montre un schéma en coupe d'une

technologie de micro-usinage en surface avant (a) et après (b) gravure des oxydes sacrificiels. On peut remarquer au passage la couche de nitrure sur tout le substrat, les trois couches structurales de polysilicium ainsi que les couches d'oxydes sacrificielles et les zones d'ancrage. La gravure des oxydes sacrificiels est généralement opérée par de l'acide fluorhydrique à 49%, suivie par une étape de séchage au CO₂ supercritique pour éviter les problèmes de collage.

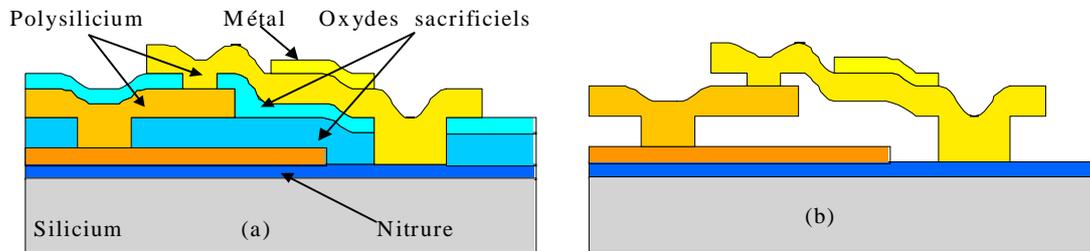


Figure 3.1. Schéma en coupe d'une technologie de micro-usinage en surface avant (a) et après (b) gravure des oxydes sacrificiels.

3.1.1.1 La micromécanique

Un des aspects intéressants des technologies de micro-usinage en surface est la possibilité de fabriquer des éléments mécaniques en 3 dimensions. On peut alors créer des systèmes mécaniques complexes comprenant des moteurs, rotatifs ou linéaires, des chaînes de transmission du mouvement comme des engrenages, des bielles et des paliers. La Figure 3.2 montre différents éléments permettant de construire des systèmes mécaniques complexes, la photo (a) représente une charnière faite avec deux niveaux de polysilicium et permettant de créer une liaison pivot, la photo (b) montre des dents d'engrenage qui ne sont pas engrenées. La photo (c) représente le détail d'un ressort rotatif.

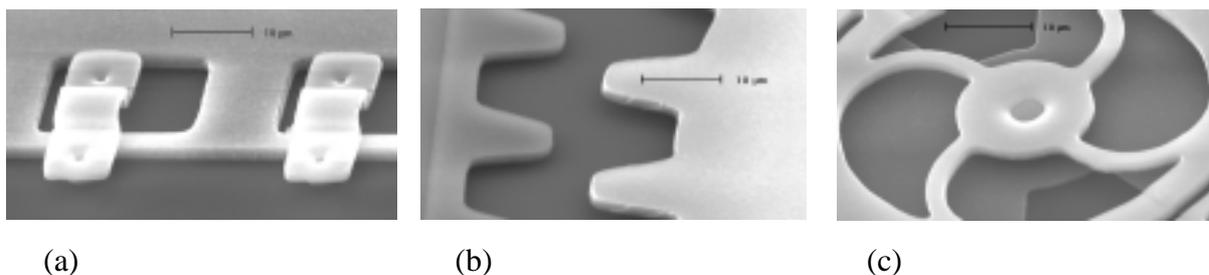


Figure 3.2. Photos MEB d'éléments micromécaniques fabriqués par micro-usinage de surface (technologie MUMPS), charnière (a), dents d'engrenage (b), ressort pivotant (c).

3.1.1.2 Les principaux fondeurs multi-utilisateurs

Il y a deux principaux fondeurs qui offrent un accès multi-utilisateurs à leur technologie. Ce sont CRONOS [10] avec la technologie MUMPS et SANDIA national labs avec les technologies SUMMIT et SUMMIT V. La technologie MUMPS est une technologie de micro-usinage en surface offrant 3 niveaux de polysilicium dont deux niveaux structurels. Le premier niveau Poly0 sert uniquement à des fins de routage électrique et d'électrodes, il n'est

pas suspendu. Les niveaux Poly1 et Poly2 avec respectivement $2\mu\text{m}$ et $0,75\mu\text{m}$ d'épaisseur forment les structures suspendues. Une couche de métal (or) peut être déposée sur le dernier niveau de polysilicium à des fins de routage faiblement résistif ou pour des applications optiques.

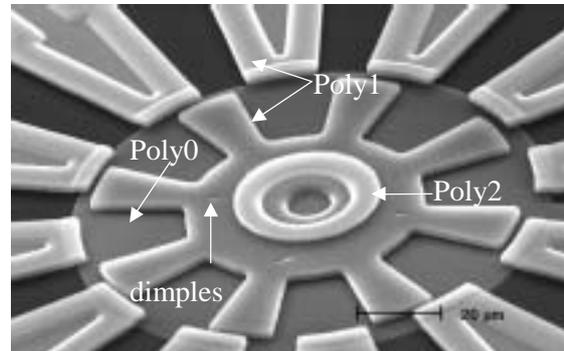


Figure 3.3. Photo MEB d'un micromoteur électrostatique fabriqué en technologie MUMPS.

Cette technologie correspond au schéma de la Figure 3.1. La photo MEB de la Figure 3.3 montre un micromoteur électrostatique fabriqué avec la technologie MUMPS, on remarque les niveaux Poly0 qui forme l'électrode de polarisation du rotor, le niveau Poly 1 qui forme le rotor mobile et les 12 électrodes fixes ainsi que le niveau Poly 2 qui est employé pour le pivot central et la bague d'arrêt verticale. Le second fondeur, les laboratoires SANDIA [11] ont un procédé microsysteme similaire à base de polysilicium et de couches d'oxydes sacrificiels contenant 3 (SUMMIT) et 5 (SUMMIT V) niveaux de polysilicium structurels. De plus ces procédés sont planarisés.

3.1.1.3 La planarisation mécano chimique

La planarisation mécano chimique est une étape technologique qui permet d'aplanir des couches d'oxydes. En effet les couches d'oxydes de silicium déposées au cours d'un process vont suivre la topologie des couches sous-jacentes. Lorsqu'un procédé comprend un grand nombre de niveaux, les décalages vont s'additionner et former un relief important. La planarisation mécano chimique va alors y remédier en polissant les couches d'oxydes déposées permettant alors de déposer les niveaux supérieurs à partir d'une surface plane. En microélectronique ces reliefs posent des problèmes notamment au niveau de la lithographie par ce que l'on appelle le *notching* qui est un rétrécissement des bandes de résine causé par des réflexions sur la topologie du rayonnement ultra violet lors de l'insolation. Pour les structures micromécaniques, la non-planéité des couches peut poser des problèmes quand des niveaux suspendus doivent glisser les uns par rapports aux autres. On remarque sur la Figure 3.4(b) qui montre une micro structure fabriquée en technologie MUMPS que le niveau supérieur de polysilicium suit les reliefs du niveau inférieur. Ces derniers vont empêcher le mouvement du niveau supérieur par rapport au niveau inférieur. Sur la photo de la Figure 3.4(a), par contre, les niveaux sont planarisés, et l'on remarque la planéité parfaite des niveaux structurels.

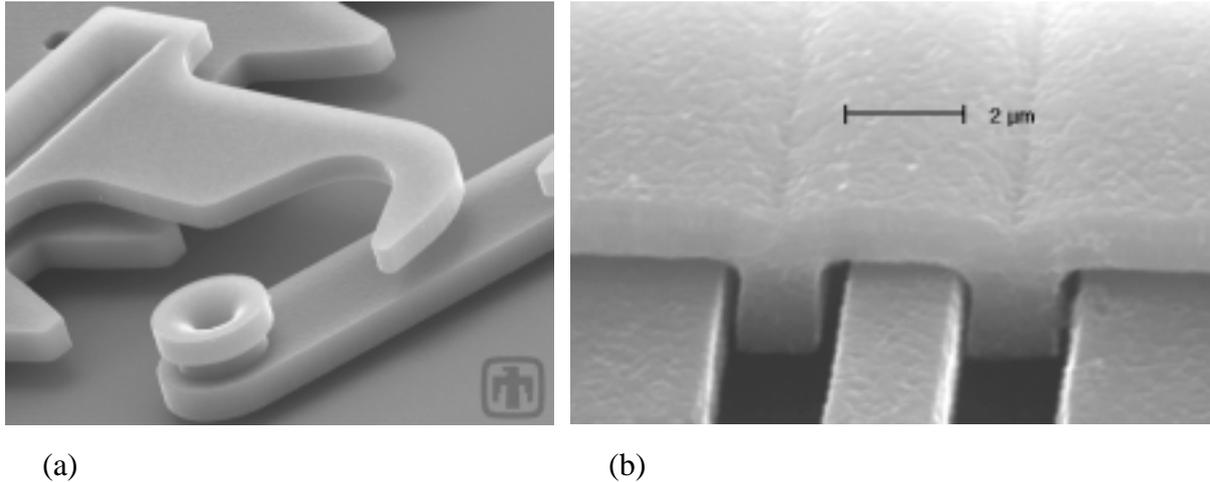


Figure 3.4. Photos MEB de structures micro-usinées en surface avec planarisation (a) (Courtesy, Sandia National Labs - MEMS, S&T Department, www.mems.sandia.gov) et sans planarisation (b).

3.1.1.4 Les problèmes de contact de surface et les dimples

La physique du contact entre surface est différente à l'échelle microscopique par rapport à l'échelle macroscopique. En effet à cette échelle les tensions de surfaces et les forces de contact sont prépondérantes par rapport aux forces gravitationnelles. C'est pour cette raison que lorsque deux microstructures entrent en contact sur une grande surface elles peuvent se coller. Pour éviter ce phénomène on trouve sur les procédés de micro-usinage en surface des niveaux de lithographie appelés "*dimples*". Les *dimples* sont des picots placés sous les structures suspendues qui évitent les contacts de surface pour minimiser les phénomènes de collage. Ces picots sont obtenus avec un niveau de lithographie supplémentaire qui va créer de petits trous dans une couche d'oxyde. Le polysilicium déposé par-dessus va alors former des petites pointes qui entreront en contact les premiers avec le niveau inférieur. On peut notamment remarquer la position des *dimples* sur le rotor du moteur électrostatique de la Figure 3.5. Le rotor va donc glisser sur le poly0 par contact entre les *dimples* et le poly0.

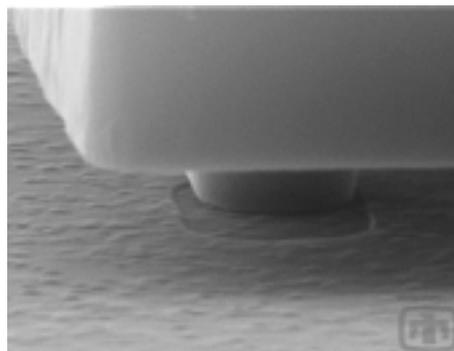


Figure 3.5. Photo MEB d'un dimple sous une couche de polysilicium suspendue (Courtesy, Sandia National Labs - MEMS, S&T Department, www.mems.sandia.gov).

3.1.1.5 Les technologies de micro-usinage de surface intégrées

Les procédés de micro-usinage de surface présentés jusqu'ici ont été élaborés à partir des techniques de dépôt en couche mince de la microélectronique spécifiquement pour fabriquer des microstructures suspendues. Or un intérêt majeur des microsystèmes est l'intégration des

fonctions de transduction et d'action avec les fonctions de traitement analogique et numérique du signal. Contrairement aux technologies de micro-usinage en volume, les procédés de micro-usinage de surface ne sont pas directement compatibles avec les procédés standard de la microélectronique et ne peuvent pas être traités en post process. Les raisons de cette incompatibilité sont toutes d'ordre thermique. En effet le dépôt des couches de polysilicium se fait à haute température et les niveaux métalliques des procédés microélectronique, le plus souvent en aluminium, ne supportent pas de telles températures. De plus un post process en température augmenterait le budget thermique d'un procédé et pourrait en modifier les profils de dopage ce qui rendrait l'électronique inopérante. Pour réaliser l'intégration il existe alors plusieurs techniques [13] [14] [16] qui ont été implémentées par différents fondeurs et que nous allons présenter ici.

❖ Procédés microélectronique et microsystemes intégrés

Cette technologie microsysteme est basée sur un procédé microélectronique CMOS modifié pour inclure des structures suspendues par micro-usinage de surface. Etant donné les contraintes de température imposées par les niveaux de métallisation, le dépôt des couches de polysilicium structurales doit intervenir avant le dépôt des couches de métal. Le procédé contient des niveaux de lithographie supplémentaires destinés à définir les zones microsystemes de manière à ce que la gravure de l'oxyde sacrificiel n'attaque pas l'électronique. La Figure 3.6 représente un schéma en coupe de cette technologie. On remarque au milieu la couche de polysilicium structurale avant gravure de l'oxyde sacrificiel. L'avantage de cette technologie est évidemment l'intégration monolithique des deux technologies mais elle est limitée quant au nombre de niveaux structurels en polysilicium.

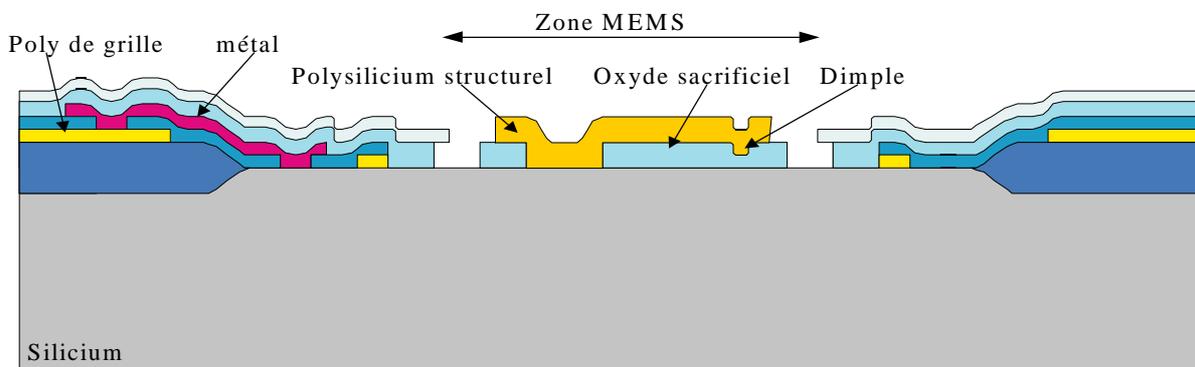


Figure 3.6. Schéma en coupe d'une technologie de micro-usinage en surface intégrée [19].

❖ Procédés microsystemes enterré : SANDIA National Labs iMEMS

La solution choisie par les laboratoires SANDIA pour intégrer les parties microsystemes à un procédé CMOS standard est de fabriquer les structures microsystemes en premier dans des tranchées obtenues par gravure anisotrope du substrat, puis d'effectuer ensuite le procédé CMOS à côté des tranchées puis finalement de libérer les structures microsystemes enterrées par gravure des oxydes sacrificiels [16]. Cette solution est réalisable parce que les structures microsystemes restent encapsulées dans les oxydes sacrificiels et qu'elles résistent aux températures élevées du procédé.

La Figure 3.7 montre un schéma en coupe d'un tel procédé, on remarque sur la partie droite la cavité contenant les structures microsystemes connectées électriquement à l'électronique CMOS fabriquée en surface et à côté. L'intérêt d'une telle solution est que le procédé

microsysteme est indépendant du procédé microélectronique et peut donc être opéré par n'importe quel fondeur et avec différents procédés. Une technologie semblable est présentée dans [15].

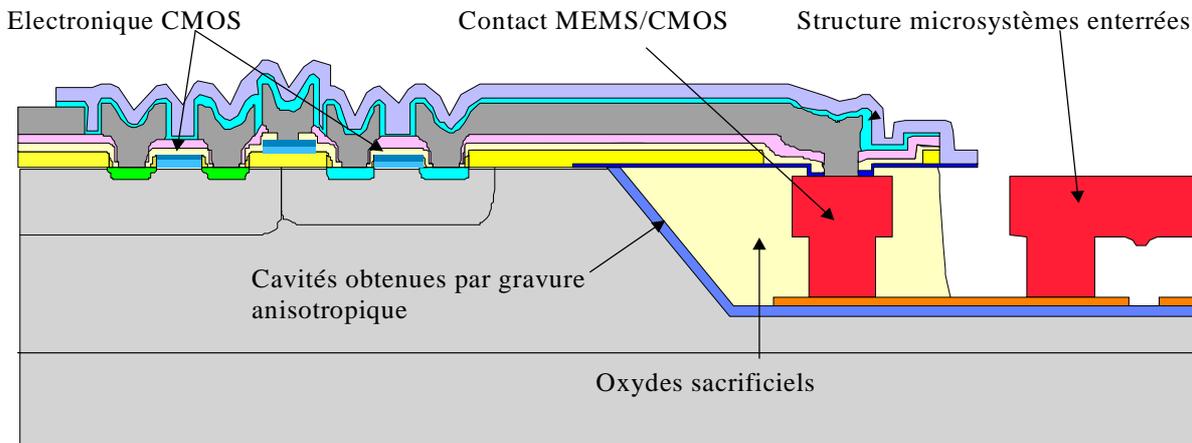


Figure 3.7. Schéma en coupe d'une technologie MEMS intégrée avec parties microsystemes enterrées, (Courtesy, Sandia National Labs - MEMS, S&T Department, www.mems.sandia.gov).

❖ Microsystemes post process

La dernière solution d'intégration monolithique de l'électronique et des microstructures en polysilicium consiste à déposer le polysilicium structural en dernier sur une technologie CMOS [12]. Cependant cela demande quelques adaptations notamment au niveau du métal des interconnexions métalliques qui sont remplacés ici par du tungstène avec des barrières de diffusion en TiN pour supporter les hautes températures de dépôt et de recuit du polysilicium des microstructures. Un dernier niveau métallique en aluminium est déposé pour les contacts après. Un des inconvénients majeur de cette solution est, outre le fait que la technologie microélectronique doit être modifiée, le niveau élevé du stress des couches de tungstène et des microstructures en polysilicium. La Figure 3.8 montre un schéma en coupe d'un microsysteme intégré avec parties microsystemes déposées après l'électronique. On remarque l'emploi de tungstène pour le premier niveau de métal et la couche de polysilicium structural après gravure de l'oxyde sacrificiel.

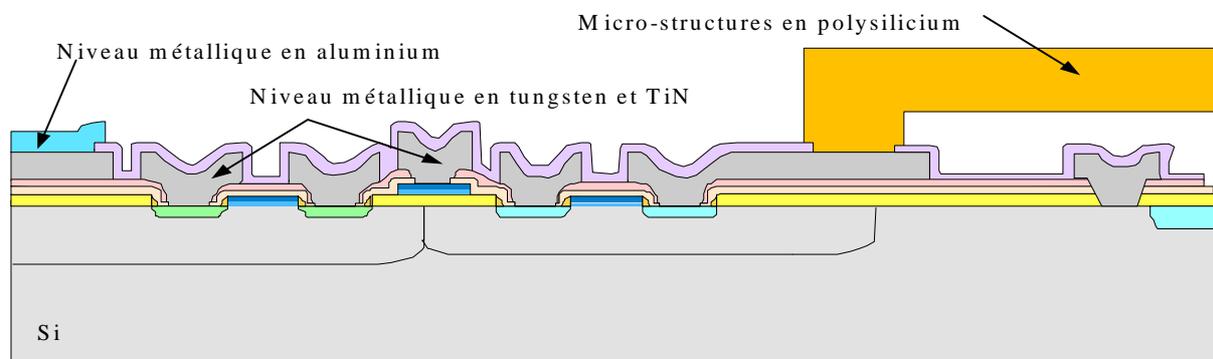


Figure 3.8. Schéma en coupe d'une technologie microsysteme intégrée avec parties déposées et gravées après l'électronique et l'usage de niveaux métalliques en tungstène.

3.1.2 Le micro-usinage en volume

Le micro-usinage en volume est une technique qui permet l'obtention de microstructures suspendues par gravure chimique du substrat d'un circuit intégré. Le terme en volume est utilisé ici puisque l'attaque chimique forme une cavité dans le volume du substrat sous les couches déposées. Cette technique est généralement utilisée directement à partir d'une technologie microélectronique, les structures suspendues étant obtenues après le post process spécifique. Dans le cas du micro-usinage en volume face avant le post process ne nécessite qu'une étape de gravure anisotrope. La définition des zones à graver se faisant simplement en empilant les ouvertures dans les différents oxydes au niveau layout de manière à laisser une zone où le matériau du substrat est à nu. La Figure 3.9 montre un schéma représentant une coupe verticale d'un process CMOS micro usiné en volume. On remarque à coté d'un transistor la cavité micro usinée laissant les couches déposées suspendues. A partir de là, la solution de gravure anisotrope va creuser dans le volume du substrat libérant des structures composées d'un sandwich des couches d'oxydes et des couches de polysilicium, de métal et de passivation. La Figure 3.10 montre une photo MEB¹³ d'une microstructure suspendue au-dessus d'une cavité obtenue par gravure anisotrope. La micro-poutre a été dessinée en décalant les différentes couches d'oxyde de manière à observer la structure en sandwich. L'extrémité de la poutre sans oxyde laisse apparaître les couches de barrières de diffusion, sous le polysilicium, ce dernier ayant été gravé par la solution de gravure anisotrope. Une des principales filières est le micro-usinage en volume face avant compatible CMOS qui permet à partir d'une technologie industrielle standard et en rajoutant un simple post process de gravure sans niveau de lithographie additionnel d'obtenir des structures microsystemes suspendues. Le post process est une gravure anisotrope du silicium du substrat réalisée avec des solutions de gravure du type KOH¹⁴ ou TMAH¹⁵.

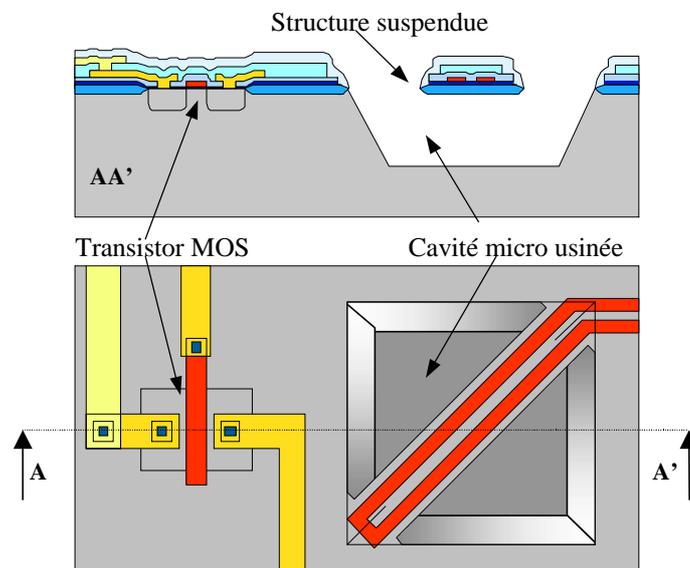


Figure 3.9. Schéma en coupe d'une technologie de micro-usinage en volume face avant compatible CMOS.

¹³ MEB microscope électronique à balayage

¹⁴ KOH hydroxyde de potassium

¹⁵ TMAH hydroxyde de tétra-méthyle ammonium

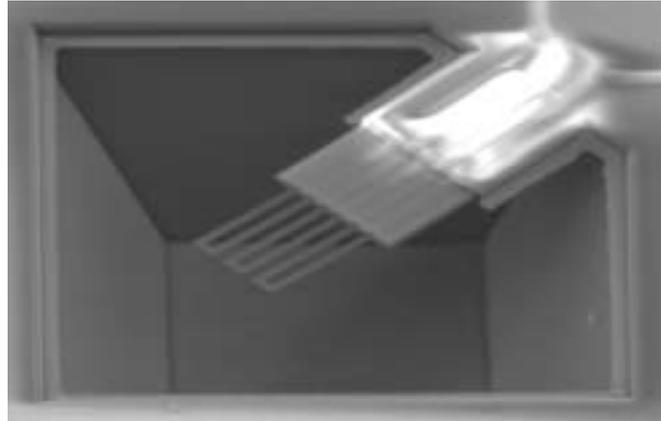


Figure 3.10. Image MEB d'une microstructure obtenue par micro-usinage en volume face avant et montrant la composition en sandwich des structures suspendues.

3.1.2.1 La gravure anisotrope du silicium

La gravure anisotrope du silicium [1] [2] est une réaction chimique en phase liquide entre la solution de gravure et le silicium et dont la cinétique est commandée par l'orientation cristallographique des surfaces de contact. En effet la vitesse de gravure exprimée le plus souvent en $\mu\text{m.mn}^{-1}$ dépend de l'orientation cristallographique de la surface en contact avec la solution. Par exemple les faces (111) du silicium ont des vitesses de gravures de plusieurs ordres de grandeurs inférieurs à ceux de plans "rapides" comme les plans (100). La Figure 3.11 montre le diagramme polaire des vitesses de gravure du silicium pour une solution de TMAH 10%, le schéma montre que la vitesse de gravure des plans (111) est très faible (mais pas nulle) par rapport aux plans (100) (surface des wafers) et aux plans (314).

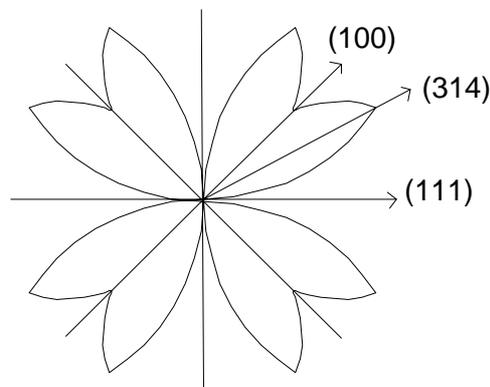


Figure 3.11. Diagramme polaire des vitesses de gravure du TMAH.

3.1.2.2 Intérêt de la gravure anisotrope

L'intérêt de la gravure anisotrope pour fabriquer des microstructures suspendues tient au fait qu'en définissant simplement une ouverture vers le silicium du substrat il est possible d'obtenir une cavité sous des structures suspendues (comme une gravure isotrope) mais également une cavité qui reste confinée par les plans d'arrêt virtuel (111) comme le ferait une gravure plasma de type RIE¹⁶. Ceci est possible du fait des très faibles vitesses de gravure des plans (111) mais également du fait des effets de bords qui permettent de graver des structures

¹⁶ RIE reactive ion etching

convexes. La Figure 3.12 montre la progression de la gravure anisotrope définie par un masque comportant deux ouvertures triangulaires orientées selon les axes CAO.

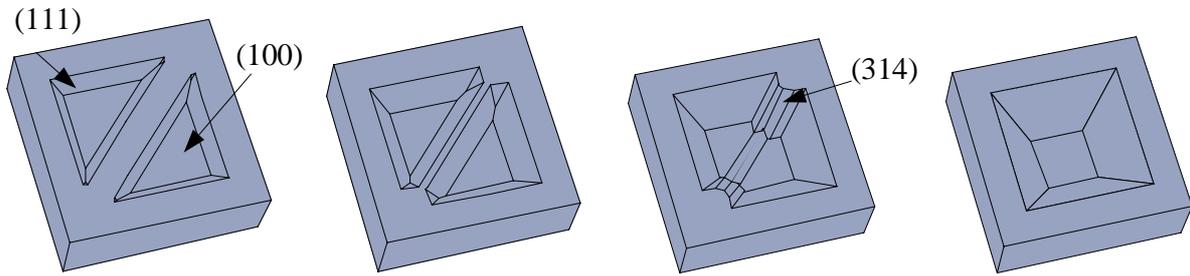


Figure 3.12. Schéma représentant l'avancée de la gravure anisotrope à partir d'ouvertures triangulaires.

On peut noter que la solution commence par consommer le plan (100) tout en formant les plans d'arrêt virtuels (111). L'effet de bord sur les parties convexes vient créer des plans (314) "rapides" permettant de libérer complètement la structure et d'obtenir une cavité en forme de pyramide inverse. Le résultat d'une telle gravure peut alors être observé sur la photo MEB de la Figure 3.13 où l'on voit une structure en pont au-dessus d'une cavité pyramidale.

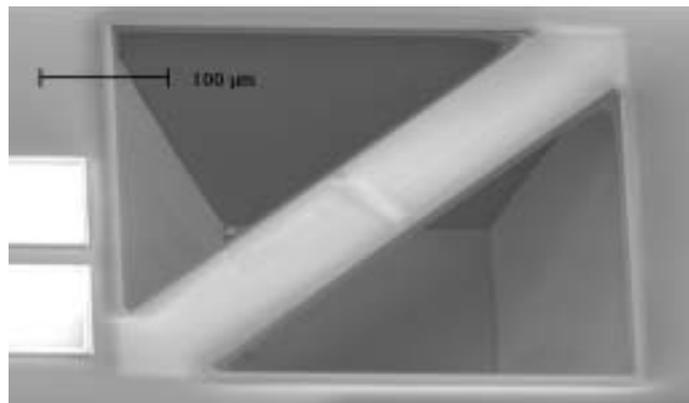


Figure 3.13. Photo MEB d'une microstructure en pont obtenu par micro-usinage en volume par gravure anisotrope du silicium du substrat.

3.1.2.3 L'anisotropie de la réaction de gravure

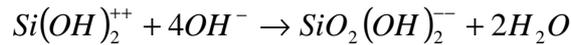
L'anisotropie d'une réaction de gravure humide d'un monocristal est un phénomène complexe. Nous n'entrerons pas ici en détail dans les phénomènes mis en jeu mais il semble que l'explication de ce caractère anisotrope vienne des différences d'orientation des hybridations orbitales sp^3 entre les différents plans cristallographiques qui créeraient des différences de vitesse de gravure. L'effet de bord ou *undercutting* qui permet l'apparition de plans rapides à l'intersection de plans (111) est également un phénomène complexe. Des publications traitent de ce sujet [3] [1] [2] [19].

3.1.2.4 Les solutions de gravure anisotrope du silicium

Les solutions de gravure du silicium présentant un caractère anisotrope sont des hydroxydes : KOH, NaOH, CeOH, RbOH, NH₄OH, TMAH. Pour cette dernière, la réaction se déroulant est alors l'oxydation du silicium par les hydroxydes pour former un silicate :



Les silicates réagissent ensuite avec les hydroxydes pour former un complexe soluble dans l'eau :



Les solutions les plus souvent utilisées pour la gravure anisotrope du silicium sont l'hydroxyde de potassium (KOH), qui offre une très grande anisotropie mais a le défaut d'être peu sélectif envers l'aluminium ce qui pose des problèmes quant aux plots de connexions dans les technologies CMOS. De plus la présence d'ions potassium qui sont des polluants des circuits CMOS est extrêmement gênante pour la compatibilité avec les procédés microélectronique. Les vitesses de gravure typiques du silicium dans une solution de KOH à 80°C sont de 1µm.mn⁻¹ et de 20 Angstrom.mn⁻¹ pour l'oxyde de silicium. L'anisotropie de gravure est de 1 pour 400 entre les plans (111) et (100). La seconde solution de gravure anisotrope du silicium est l'éthylène diamine pyrocathécol (EDP) qui bien qu'offrant de meilleures caractéristiques de sélectivité reste peu employée du fait de sa très haute toxicité et de son interdiction en salle blanche. La vitesse de gravure typique du (100) du silicium dans une solution EDP à 80°C est de 20µm.mn⁻¹ ce qui est beaucoup plus grand que le KOH, l'anisotropie est de 1 pour 35 entre les plans (111) et (100). La dernière solution de gravure est l'hydroxyde de tétra-méthyle ammonium (TMAH) qui bien que d'un emploi délicat du fait de sa nature organique qui lui confère une faible stabilité, reste la solution offrant les meilleurs résultats en termes de vitesse de gravure, d'anisotropie et de sélectivité par rapport à l'aluminium. De plus elle est totalement compatible avec les procédés de fabrication de la microélectronique. Une solution de gravure à base de TMAH a été développée [4] au laboratoire TIMA en collaboration avec la société IBS et est utilisée comme post process des technologies bulk micromachining au CMP. Dans la suite de cet exposé la majeure partie des microsystemes présentés ont été gravés avec cette solution.

3.1.2.5 Les différents types de micro-usinage en volume

Le micro-usinage en volume face avant compatible CMOS [9] permet d'obtenir facilement des structures suspendues sans étapes de lithographies supplémentaires. La Figure 3.14 montre les différentes structures qu'il est possible d'obtenir, une structure suspendue en forme de pont (a), de poutre (b) ou de membranes suspendues par des bras de support (c).

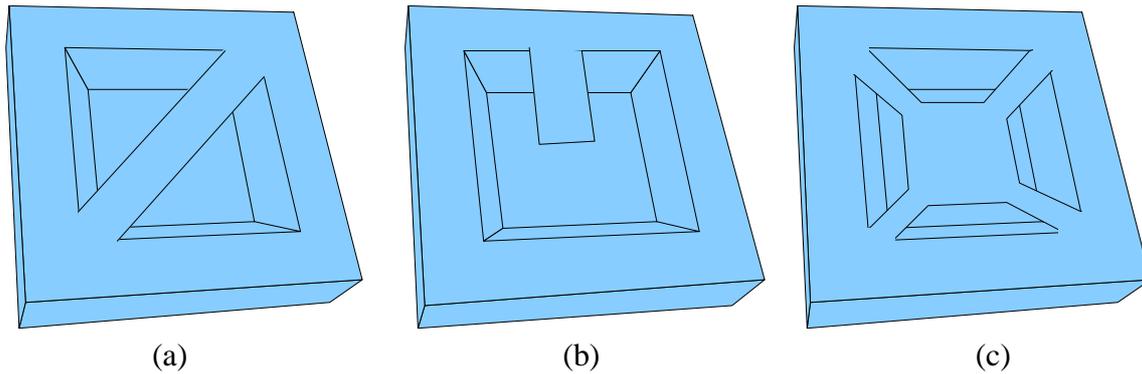


Figure 3.14. Schéma des différentes microstructures que l'on peut obtenir par micro-usinage en volume face avant.

En opérant une gravure assez longue, il est possible aussi de traverser l'épaisseur du wafer de manière à laisser une masse suspendue à une membrane [4]. La gravure s'opérant également par la face arrière du wafer (dans le cas d'une gravure au niveau puce). Néanmoins cette technique impose une taille assez grande de la structure suspendue (typiquement $800\mu\text{m}$ de côté). La Figure 3.15 illustre cette technique et montre une membrane suspendue par quatre bras de support et à laquelle est accrochée une masse en silicium de forme tronconique. On remarque au passage que la forme coudée des bras de support est une méthode pour retarder la gravure de la masse centrale par les effets d'arêtes au moment de la gravure anisotrope. De fait la masse suspendue n'est pas tout à fait pyramidale mais arrondie au niveau des arrêtes. Une masse suspendue de ce type peut alors faire office de masse sismique dans des applications de capteurs inertiels ou de masse thermique.

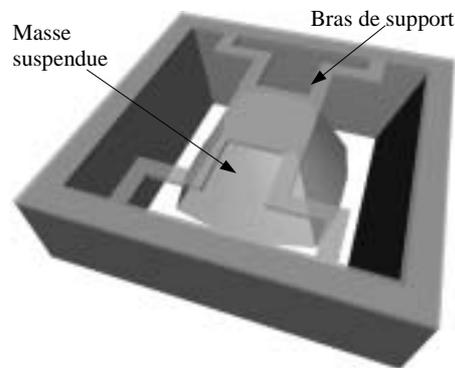


Figure 3.15. Schéma d'une structure à masse suspendue obtenue en micro-usinage en volume face avant [19].

3.1.2.6 Le micro-usinage en volume face arrière

Le micro-usinage en volume face arrière nécessite quant à lui une étape supplémentaire de lithographie de manière à créer un masque pour la gravure anisotrope. Cette étape de lithographie doit être alignée par rapport à la face avant, ce qui nécessite l'emploi d'un aligneur double face. De plus cette technique impose de travailler au niveau wafer et non au niveau puce comme c'était le cas pour le micro-usinage de surface en face avant. La Figure 3.16 montre les structures qu'il est possible d'implémenter avec cette technique. La Figure 3.16(a) montre une membrane close obtenue par gravure en face arrière, c'est cette technique qui est employée dans les capteurs de pressions ou la déflexion de la membrane suite à une

différence de pression entre ses deux faces est mesurée. Notons aussi que la cavité formée sous la membrane peut être fermée par collage du dispositif sur une surface de silicium (*wafer bonding*) et mettre ainsi une référence de pression dans la cavité, on a alors un capteur de pression absolue. La Figure 3.16(b) montre une membrane close à laquelle est attachée une masse suspendue, pour des applications thermiques ou de capteurs inertiels.

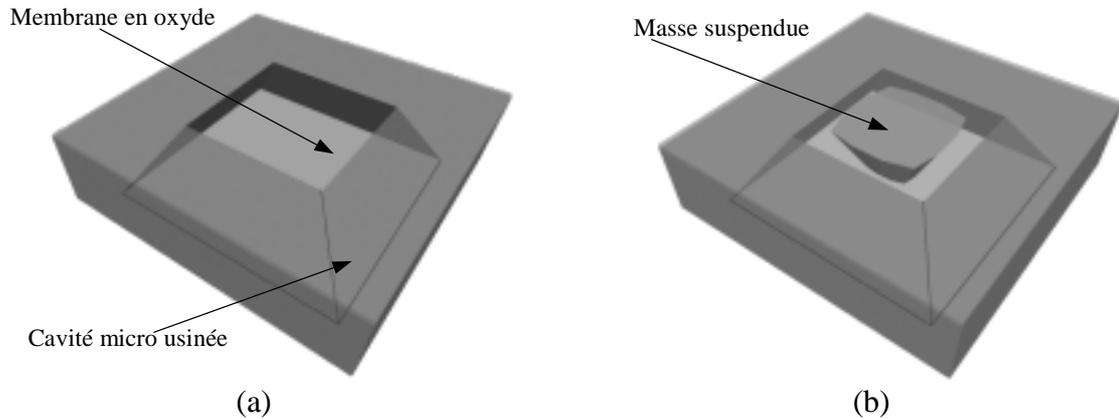


Figure 3.16. Schéma des différentes structures résultantes d'un micro-usinage en volume face arrière.

3.1.2.7 Le micro-usinage en volume face avant et arrière

En combinant le micro-usinage en volume face avant et face arrière, il est possible d'obtenir des structures suspendues avec masse attachée. L'intérêt de cette méthode est que la gravure va s'opérer des deux cotés en même temps ce qui va diminuer le temps de gravure nécessaire pour suspendre complètement une masse de silicium. De plus la taille minimum des ouvertures à pratiquer sera plus petite. La Figure 3.17 montre un schéma en coupe d'une structure à masse suspendue analogue à celle présentée en Figure 3.15 mais cette fois ci obtenue en micro-usinage en volume face avant et face arrière. On remarque la forme en diamant de la masse suspendue et la superposition dans le plan des masques de gravure avant et arrière.

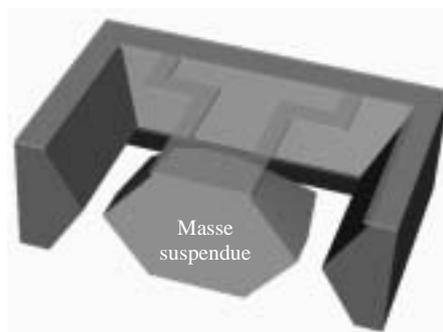


Figure 3.17. Schéma en coupe d'une structure à masse suspendue obtenue en micro-usinage en face avant et en face arrière [19].

3.1.2.8 Le micro-usinage en volume électrochimique

Cette technique de gravure exploite l'effet de la polarisation électrique d'un matériau sur les vitesses de réaction de gravure. Dans le cas du micro-usinage en volume, la gravure électrochimique [7] permet d'obtenir des couches d'arrêt de la gravure simplement en polarisant les caissons (n uniquement) implantés dans le silicium. La gravure anisotrope

s'opérant en phase liquide, on doit prendre une référence de potentiel dans la solution. Il est possible alors d'obtenir des structures à membrane épaisse formées par les zones implantées n non gravées comme le montre la Figure 3.18. Une autre possibilité est de suspendre des zones de silicium isolées contenant de l'électronique (uniquement Pmos) qui serait isolée électriquement et thermiquement du substrat.

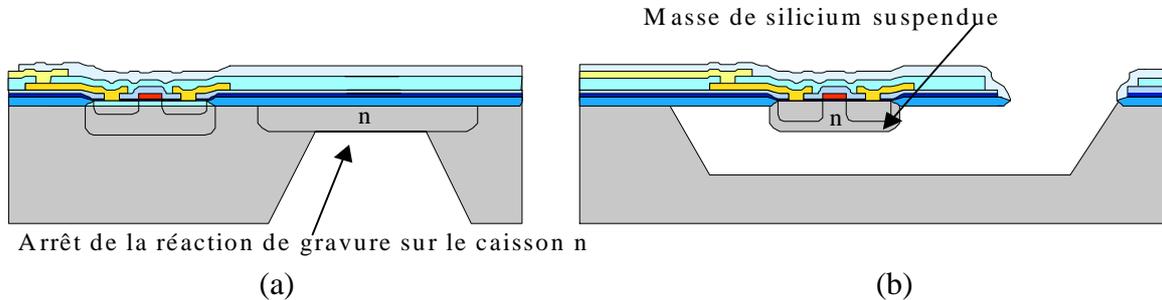


Figure 3.18. Schéma en coupe de structures suspendues par micro-usinage électrochimique en volume, face arrière (a) et face avant (b).

3.1.3 D'autres technologies

Il existe beaucoup d'autres technologies de micro fabrication, nous les décrivons rapidement ici car très peu de ces technologies sont utilisées pour faire des systèmes intégrés.

3.1.3.1 Les microsystemes sur SOI

Le SOI¹⁷ est une technologie microélectronique CMOS qui permet d'obtenir de meilleures performances en hautes fréquences, en tenue aux radiations, en température et une consommation réduite par rapport aux technologies CMOS classiques. Le principe est de réaliser un procédé CMOS sur un wafer contenant une couche d'oxyde enterrée sous le silicium. En ce qui concerne les microsystemes cette technologie permet d'obtenir des structures mobiles en silicium monocristallin après gravure RIE¹⁸ du silicium monocristallin et gravure isotrope de la couche d'oxyde enterrée. Cette dernière agissant comme couche d'oxyde sacrificiel, la Figure 3.19 montre un schéma en coupe de cette technologie.

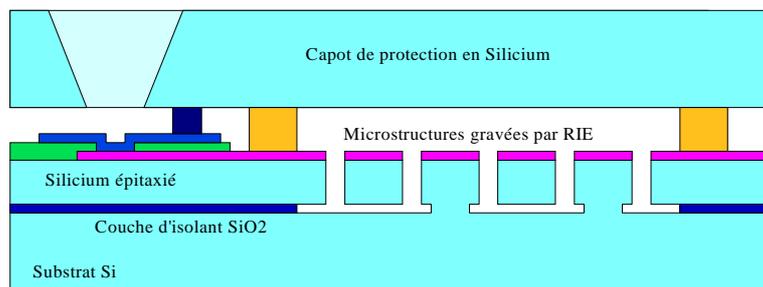


Figure 3.19. Schéma en coupe de la technologie de micro-usinage sur SOI : Epi-SOI de Tronic's microsystems (Courtesy of TRONIC'S Microsystems, France).

¹⁷ SOI *Silicon On Insulator*, silicium sur isolant

¹⁸ RIE *reactive ion etching*, gravure par plasma

3.1.3.2 Le LIGA

Le LIGA¹⁹ [18] est une technologie d'origine allemande qui utilise la croissance électrolytique de métaux. La Figure 3.20 montre un schéma du procédé LIGA. Une couche de photoresine épaisse est déposée sur un substrat, une étape de lithographie aux rayons X ou UV est opérée. La croissance du métal se fait alors dans la forme de résine. Une couche de précurseur (*seed*) métallique est nécessaire sur le substrat pour démarrer la réaction de croissance électrolytique. La résine est ensuite nettoyée. Ce procédé permet d'obtenir directement des pièces métalliques ou bien des moules métalliques pour réaliser des microstructures en plastique par injection.

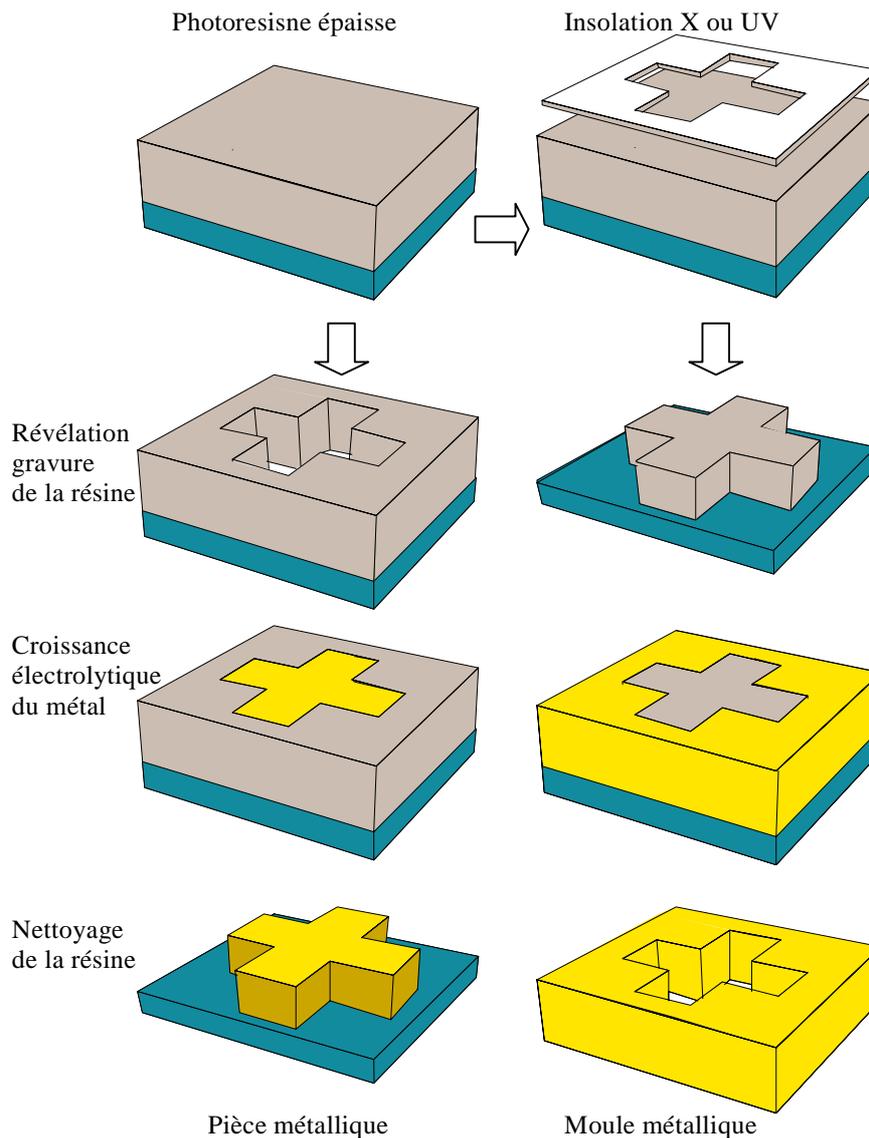


Figure 3.20. Procédé LIGA pour la fabrication de pièces métalliques ou de moules métalliques pour injection de plastiques [19].

¹⁹ LIGA pour Litographie Galvanoformung Abformung

3.2 Simulation de procédé technologique et génération de modèle 3D d'une structure microsysteme

Nous avons implémenté un outil appelé "3-D Solid modeler". Cet outil génère un modèle 3-D d'un layout de microsysteme en interprétant des informations de masque (GDSII, ou CIF) et les paramètres de la technologie utilisée.

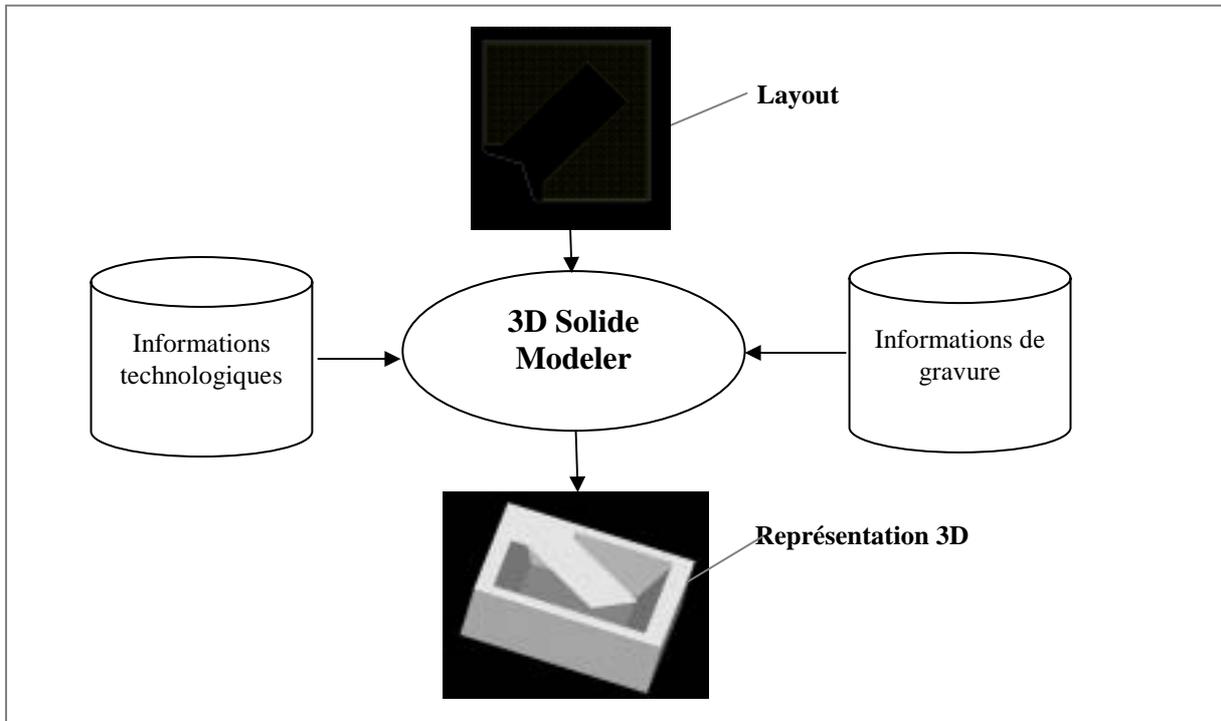


Figure 3.21. Le flot dans un environnement de modélisation 3D

Le 3D Modeler émule les effets géométriques du processus de la fabrication sur un wafer à partir de son layout et des informations technologiques du procédé.

Le 3D Modeler pourrait être utilisé pour capturer des erreurs dans le layout où dans le processus de fabrication avant de soumettre le design à la fonderie. Il présente aussi une étape principale pour la génération des fichiers d'entrée pour les outils d'analyse 3D comme ceux à base d'éléments finis.

La modélisation 3D peut être faite de deux manières différentes :

- La première méthode consiste à construire les structures microsysteme directement en 3D, c'est la méthode adoptée dans le logiciel HFSS de ANSOFT. On peut reprocher à cette méthode d'être difficile à manœuvrer puisqu'elle ne respecte pas le flot microsysteme. On peut ajouter aussi que c'est une méthode coûteuse au niveau du temps et du calcul.
- Dans la deuxième méthode, il s'agit de partir d'un dessin de masque microsysteme représenté sous une forme standard (GDSII, CIF) et d'une base de données qui contient d'une part les informations technologiques du procédé utilisé (essentiellement les épaisseurs des couches), et d'autre part la définition du procédé. C'est la méthode qui est adoptée ici.

3.2.1 Définition du procédé

L'élément principal de cette méthode est le procédé technologique. Le procédé est une séquence d'opérations de fabrication qui varient principalement entre la déposition ou la suppression de la matière (gravure) suivant un masque donné. Dans les sections suivantes, on va détailler les différents types de chacune de ces opérations :

3.2.1.1 Déposition d'une couche

Il y a plusieurs méthodes pour déposer une couche sur une autre pendant un procédé technologique, ces méthodes varient d'un procédé un autre. En général on peut citer quatre types de déposition :

I. Déposition « Conformal » :

C'est le type de déposition le plus courant dans les procédé technologique. Ce type est illustré dans la Figure 3.22 :

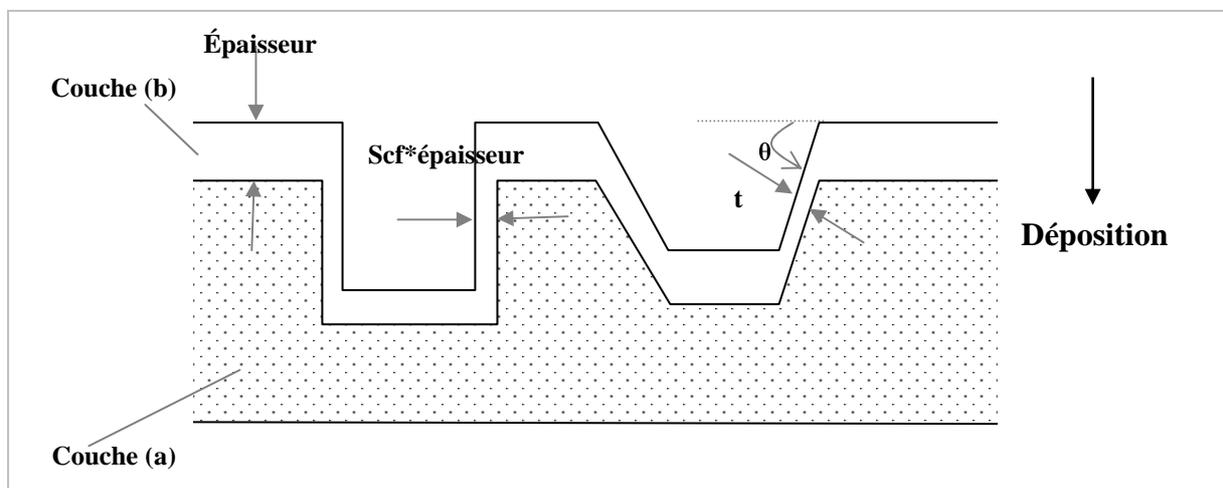


Figure 3.22. Déposition « conformal » d'une couche sur une autre.

Une déposition « conformal » est paramétrée par 3 variables :

- **Épaisseur** : l'épaisseur de la matière déposée pour la dimension verticale.
- **Scf** : (Side wall coverage factor). Ce paramètre est la hauteur de la matière déposée sur les cotés verticaux divisés par l'épaisseur de la matière sur les faces horizontales.
- **t** : (Material coverage) sur les faces inclinées dépend de l'angle d'inclinaison de la face, ce paramètre est donné par la relation : $t = \sqrt{\sin^2(\theta) + Scf^2 \sin^2(\theta)}$

Il faut noter ici que dans ce type de déposition toutes les faces de la couche (a) sont couvertes par la couche (b) même si ce sont des faces ombrées comme illustré dans la Figure 3.23.

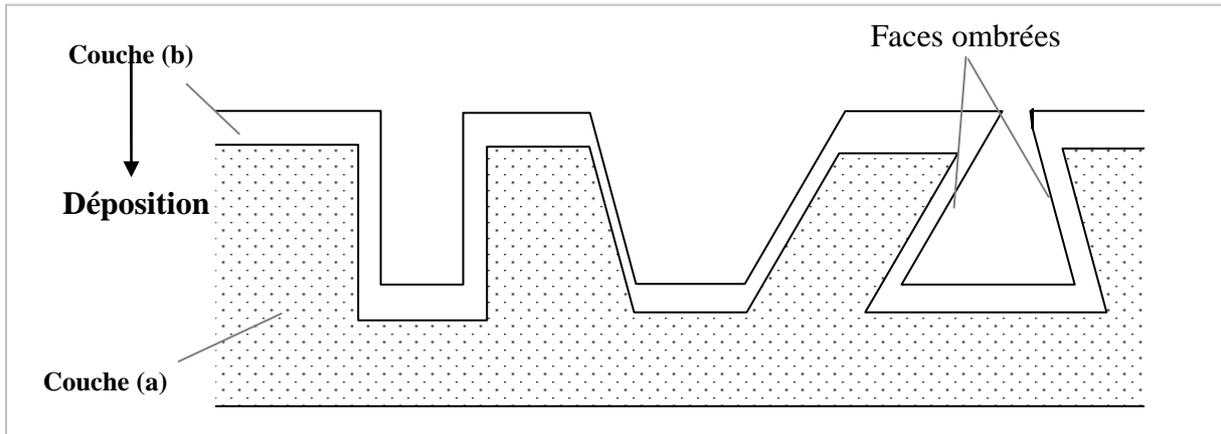


Figure 3.23. Déposition sur les faces ombrées.

II. Déposition « Snowfall »

Dans ce type de déposition, les faces horizontales de la couche (a) sont couvertes avec maximum d'épaisseur de la couche (b). Il n'y a pas de déposition sur les faces verticales ni sur les faces ombrées. Les faces inclinées qui ne sont pas ombrées sont couvertes avec un minimum d'épaisseur de la couche (b) comme il est montré dans la Figure 3.24.

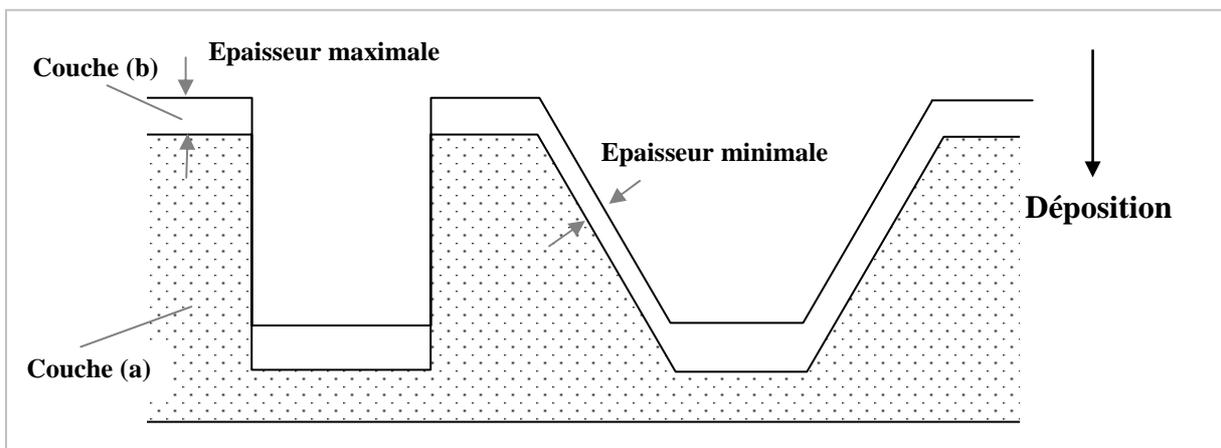


Figure 3.24. Déposition « Snowfall ».

Ce type de déposition est paramétré par deux variables qui sont le maximum et le minimum de l'épaisseur de déposition.

III. Déposition de remplissage « Fill »

Ce type de déposition est le plus simple, cette déposition consiste à remplir la topologie courante avec la couche déposée, comme illustré dans la Figure 3.25 :

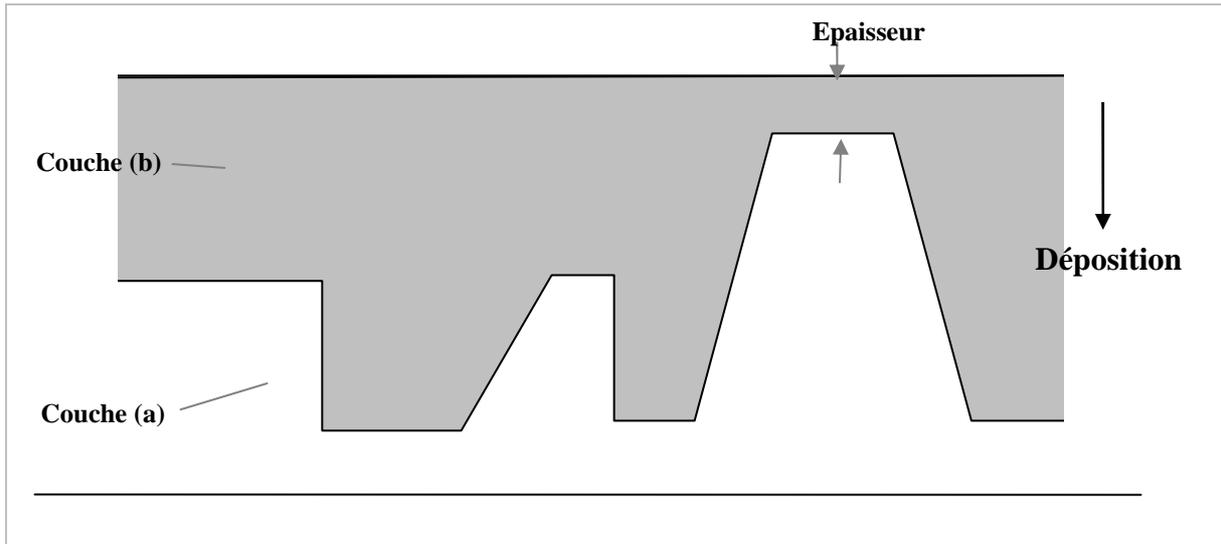


Figure 3.25. Déposition «Fill».

Il faut noter que le seul paramètre d'une telle déposition est la variable épaisseur qui présente la hauteur de la couche (b) par rapport à la face la plus haute de la couche (a).

3.2.1.2 Suppression d'une couche (gravure)

Comme nous l'avons expliqué auparavant, il y a différents types de gravure qui varient selon la technologie utilisée. Dans la section suivante, on va expliquer les différents paramètres de chaque type de gravure pris en considération dans le 3D Modeler.

I. Gravure en surface

Dans ce type de gravure, on supprime les couches qui sont déposées pendant les étapes précédentes suivant un masque de gravure, on peut définir deux types de masque de gravure :

Masque de gravure inclusif : la gravure s'effectue au dessous du masque de gravure comme illustré dans la Figure 3.26.

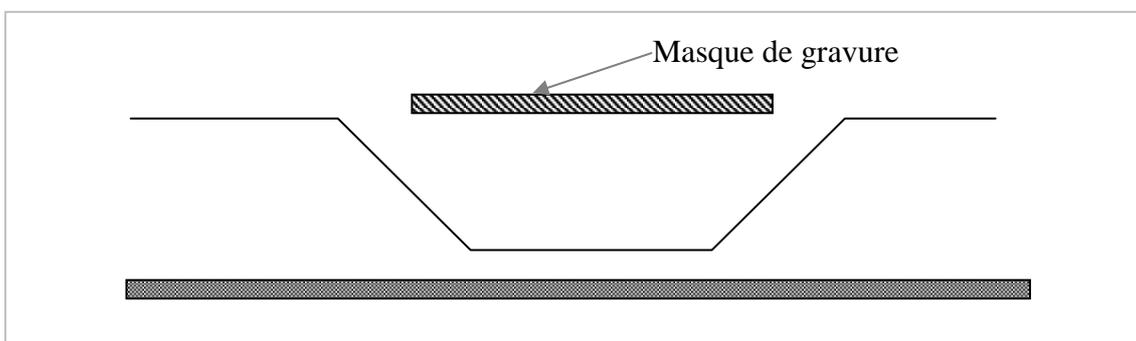


Figure 3.26. Gravure en surface inclusive.

Masque de gravure exclusif : les zones qui se trouvent au dessous de ce type de masque restent protégées, et la gravure s'effectue partout ailleurs.

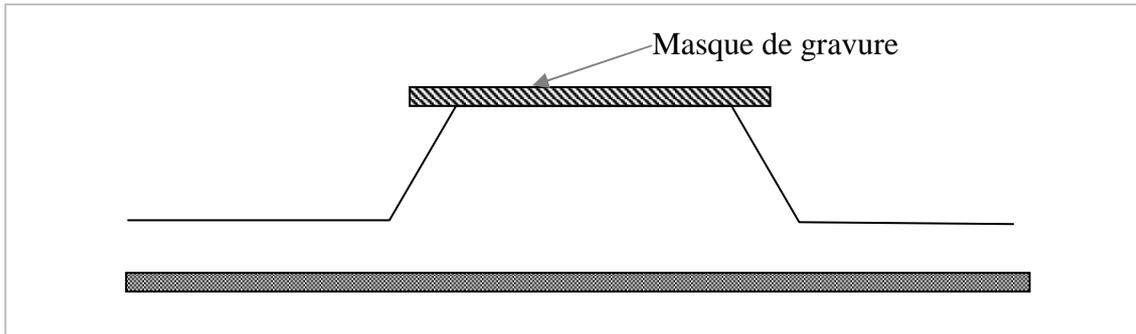


Figure 3.27. Gravure en surface exclusive.

Les différents paramètres de gravure en surface sont montrés dans la Figure 3.28.

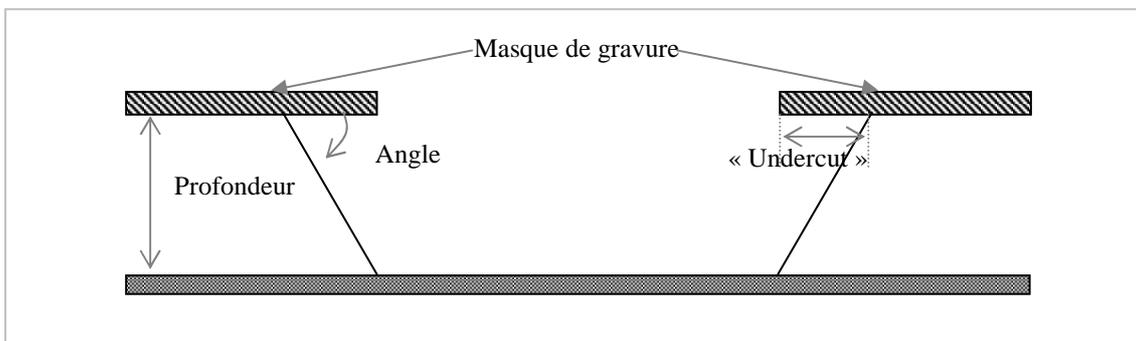


Figure 3.28. Les paramètres de gravure en surface.

Il y a trois paramètres décisifs dans la gravure en surface :

- La profondeur de la gravure.
- L'angle d'inclinaison des faces produites par la gravure.
- L'avancement de gravure par rapport au bord intérieur du masque de gravure.

II. Gravure en volume

La simulation de la gravure anisotrope peut s'effectuer selon deux approches distinctes, la simulation atomistique et la simulation géométrique.

Dans la première méthode (atomistique), la simulation est basée sur la discrétisation d'un volume en points (les atomes) d'un réseau régulier, comme un réseau d'atomes dans un cristal. L'algorithme [5] va alors, pas à pas, calculer l'orientation du plan formé par un point et ses proches voisins puis en extraire une vitesse de gravure correspondante. En fonction du pas de calcul et de la vitesse de gravure déterminée, l'atome sera alors déclaré gravé ou pas. La Figure 3.29 montre une simulation atomistique en 3 dimensions de la progression de la gravure anisotrope d'une structure en forme de membrane soutenue par quatre bras à 45°. Cette simulation a été effectuée avec le simulateur de gravure anisotrope ACES [6].

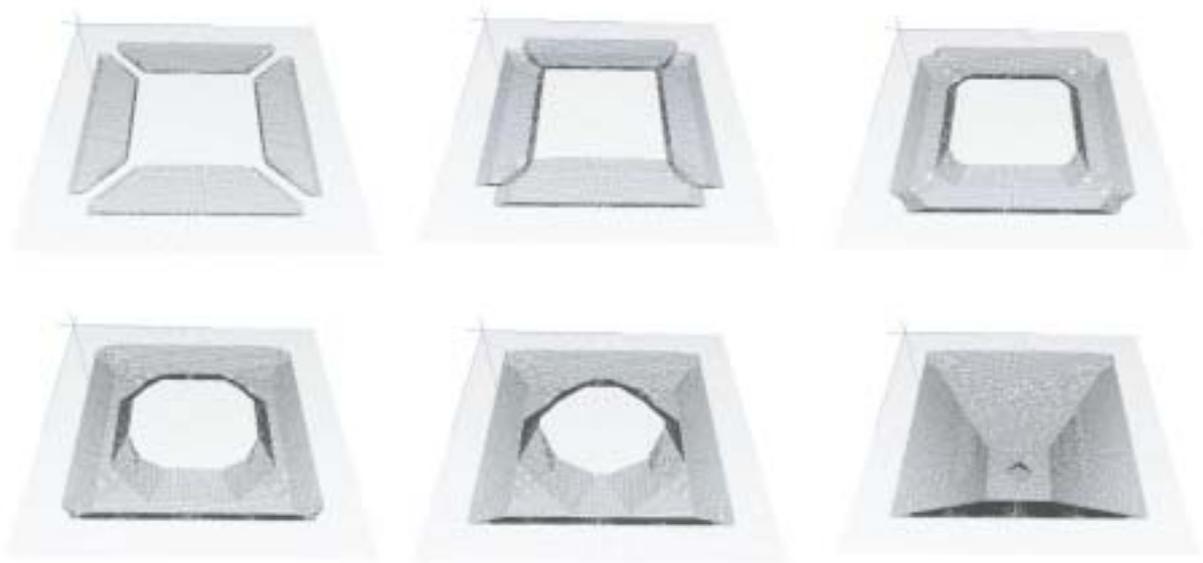


Figure 3.29. Simulation atomistique de la gravure anisotrope d'une membrane soutenue par quatre bras à 45°, pour différents pas de calculs montrant la progression de la gravure.

Dans le deuxième méthode (géométrique), la progression de la gravure au cours du temps est représentée par l'évolution de vertex définissant l'intersection de plusieurs plans. En fonction des vitesses de gravure de ces plans la position de vertex évolue et peut mener à la disparition ou à l'apparition de nouveaux plans. Plusieurs algorithmes de simulation géométrique de la gravure anisotrope existent et ont été utilisés pour construire des simulateurs.

Dans notre approche, l'approche géométrique est l'approche utilisée, cette approche ne nécessite que deux données pour l'étude de gravure :

1. Le diagramme des vitesses :

Le diagramme de vitesse ou *Each Rate Diagram* est un diagramme en coordonnées polaires qui caractérise le produit gravant [19]. Il représente les vitesses des plans de Miller en Fonction de l'angle qu'ils font avec le plan de référence (généralement, on prend comme plan 100 pour des substrats Si dopé p). La symétrie des cristaux fait qu'on a un diagramme symétrique.

2. Le masque ou la structure à graver.

La gravure s'effectue soit sur des formes tridimensionnelles, soit sur une plaque de semi-conducteur masqué. Pour graver une forme tridimensionnelle, il est nécessaire d'avoir une base théorique solide de gravure 3D, nous nous contentons dans notre approche de donner un aperçu sur la théorie bidimensionnelle puisque c'est l'approche utilisée.

i La théorie géométrique de la gravure bidimensionnelle

La théorie géométrique de la gravure bidimensionnelle se base sur la connaissance des vitesses des plans de Miller [19]. Le masque a au début une forme géométrique, cette forme est constituée de segments de droites (faces) reliées entre elles par des sommets. Cette approche consiste à attribuer à chaque segment la vitesse équivalente prise dans le

diagramme de vitesse. On répertorie trois événements : les créations de faces, les disparitions de faces et les intersections de faces [19].

Les faces se déplacent suivant le vecteur vitesse normal associé. Dans ce déplacement, leur longueur diminue ou augmente. Dans le premier cas, et à un instant donné, elles disparaissent. Il arrive aussi que deux ou plusieurs faces interagissent, et que d'autres apparaissent entre deux faces.

- **Apparition des faces :**

Les coins concaves et les coins convexes du masque ne possèdent pas les mêmes propriétés. Dans un coin concave, pour qu'une face apparaisse, il faut que son vecteur vitesse ne coupe pas la corde reliant les sommets des vecteurs vitesses des deux faces voisines (théorie des « *convex & concave hull* ») [20]. Dans le cas d'un coin convexe, il suffit que le vecteur vitesse soit plus grand.

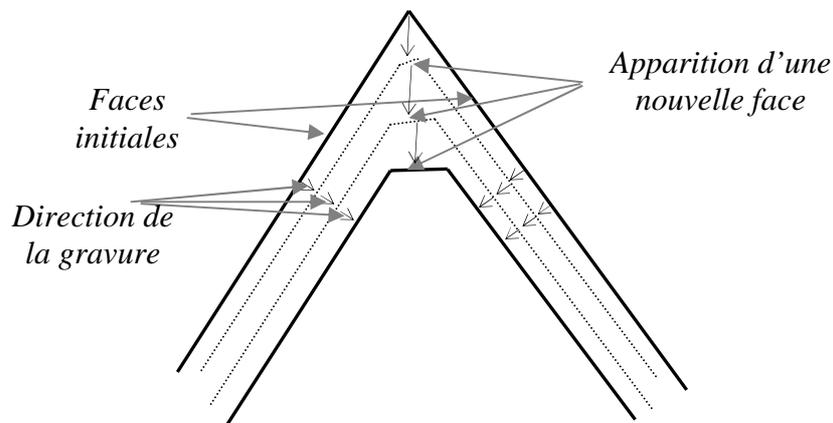


Figure 3.30. Apparition d'une nouvelle face.

Il arrive que l'on rencontre des cas où plusieurs faces sont candidates pour apparaître. On prend alors les faces qui vérifient le critère « *convex & concave hull* » par rapport aux deux faces candidates voisines [20].

Le critère « *convex & concave hull* » est facile à démontrer. En effet, dans le cas des faces concaves, si une face apparaît, pour qu'elle ne soit pas détruite par ses voisines, il faut que la vitesse soit plus faible car sinon la trajectoire de ses deux points limites converge. La vitesse d'un point de contact de deux faces est calculée en fonction des vitesses de ces faces. Il est perpendiculaire à la corde reliant les extrémités de ces vitesses.

- **Disparition de faces**

C'est le processus inverse des apparitions de faces. Les règles qui gouvernent cet événement sont similaires à celles de la section précédente. Quand les deux vecteurs vitesse des points limites d'une face convergente, la face disparaît au bout d'un certain temps que l'on peut calculer à partir du diagramme de vitesses.

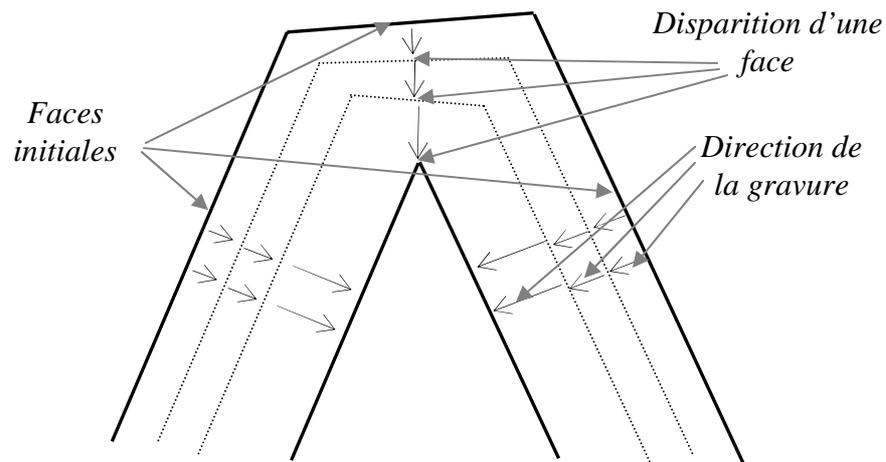


Figure 3.31. Disparition d'une face.

- **Intersection des faces**

Cet événement se produit si deux ou plusieurs faces émergent ensemble vers un seul point, on peut distinguer deux cas d'intersections, l'intersection des faces parallèles (Figure 3.32) et l'intersection des faces non-parallèles (Figure 3.33).

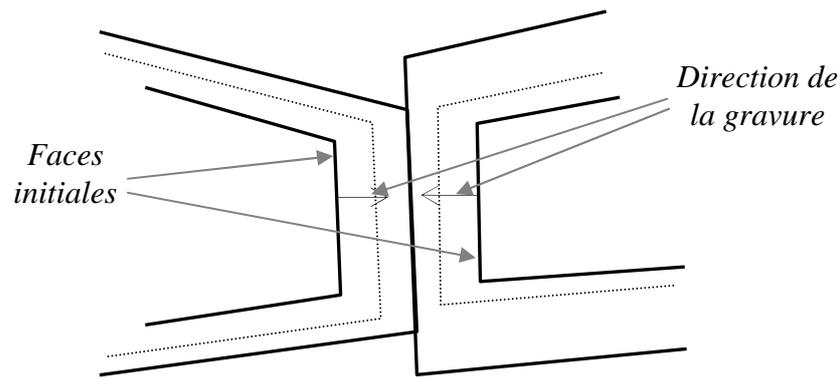


Figure 3.32. Intersection de deux faces parallèles.

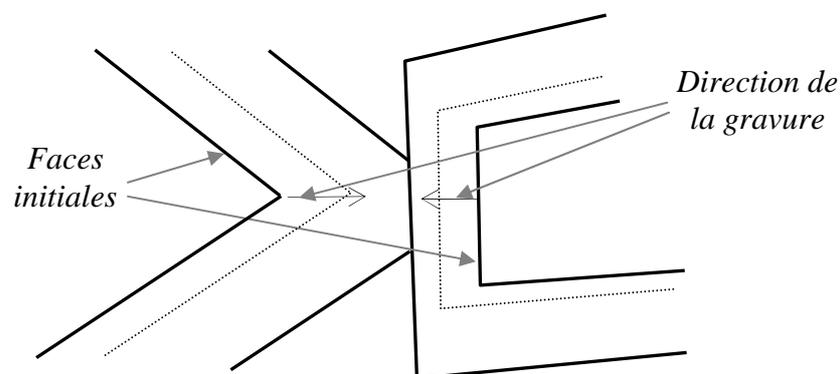


Figure 3.33. Intersection des faces non parallèles.

ii L'algorithme de simulation de la gravure bidimensionnelle

L'algorithme de la simulation de la gravure est contrôlé par trois fonctions principales :

• **La fonction de convergence des faces :**

Cette fonction détermine la convergence ou la non convergence d'une face donnée. Cette fonction est calculée à partir de la vitesse des 2 points qui limitent la face traitée. Plusieurs configurations définies par l'orientation des vecteurs vitesses sont prises en considération comme il est illustré dans la Figure 3.34.

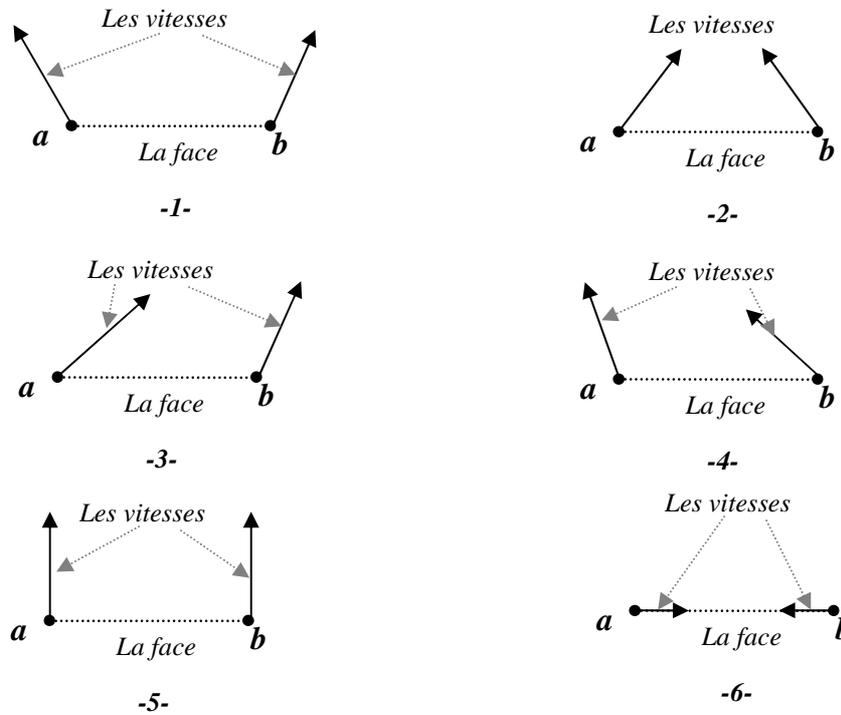


Figure 3.34. Les différentes configurations de vitesses.

Ces configurations contrôlent les apparitions, les disparitions et les intersections des faces.

• **La fonction de la disparition des faces :**

Nous avons créé un tableau de structures de disparition. Chaque structure de disparition possède deux champs : le premier champ indique le temps restant à la disparition de la face si cette dernière converge et le deuxième champ identifie de quelle face il s'agit.

Le tableau de structure de disparition est par ailleurs classé dans l'ordre des temps de disparition croissants. Ainsi, à l'instant présent, seules les faces dont les temps de disparition sont relativement proches disparaissent. De plus, à chaque événement, le tableau de structure est scruté afin d'en éliminer les faces qui auraient été effacées du tableau des faces durant la gestion de l'événement précédent.

Lorsque le prochain événement est une disparition, pour chaque face existante, on vérifie si la face converge ou non. Si elle converge, on calcule sa durée de vie à partir du temps courant, estimée à partir de la vitesse de la face et de sa longueur.

Lorsque le prochain événement est une intersection, on vide complètement le tableau de structures de disparition, afin que tous les temps de disparition soient recalculés et remis dans le tableau de structures. Ceci est fait afin d'éviter des problèmes avec des apparitions ou des disparitions de faces pendant la gestion des intersections.

Dans les deux cas, c'est-à-dire si le prochain événement est une disparition ou des intersections, on soustrait la valeur de chaque temps stocké dans le tableau de structure de disparition avec la valeur de pas de temps. Ainsi, à chaque nouvel événement, on connaît les nouveaux delta-temps qui séparent l'instant présent et les disparitions à venir.

- **La fonction d'apparition des faces :**

Nous avons placé un test avant chaque appel de fonction d'apparition de faces, ce test vérifie premièrement si la face candidate à l'apparition est plus petite qu'une valeur réelle choisie de telle sorte que l'approximation de la simulation soit la plus juste possible, et ensuite si les deux faces entre les quelles la ou les nouvelles faces apparaissent existent encore.

L'algorithme de la simulation de gravure bidimensionnelle est :

```

Lecture des fichiers d'entrées
Initialisation des données
N ← 1
{ -- Boucle Principale -- }
Tant que (fin de simulation = faux) faire
  début
    Pour chaque face faire
      Début
        Si Convergence (face) = vrai
          Disparition (face).
        Sinon
          Apparition (faces).
          Mettre à jour tableau de structures.
          En déduire le prochain événement et le Δt associé (Δt = temps
          de l'événement - temps courant).
        Fin
      Si (temps courant + Δt >= N * Temps de simulation / nombre de pas) faire
        Début
          Déplacer les points suivants leurs vitesse vers le temps
          (N * Temps de simulation / nombre de pas).
          Sauvegarder l'état de la structure.
          Incréments N et calculer nouveau Δt.
          Déplacer les points suivants leur vitesse et le Δt .
          Incréments le temps courant.
          Gérer l'événement (apparition ou disparition).
        Fin
      Sinon
        Début
          Déplacer les points suivants leur vitesse et le Δt .
          Incréments le temps courant
          Gérer l'événement (apparition ou disparition).
        Fin
    Fin { --de boucle principale -- }

```

Algorithme 3.1. L'algorithme de simulation de gravure.

iii Implémentation, simulation et résultats

Le premier simulateur de gravure implémenté avait pour but de prévoir si une structure sera suspendue avec un produit gravant spécifié, et au bout de combien de temps. Il était nécessaire d'implémenter une première version « *Stand alone* ». Cette version a été réalisée pour tourner sous système UNIX. L'affichage des résultats est sous le logiciel Gnuplot, la Figure 3.35 montre les résultats de deux simulations de gravure d'une poutre avec deux pas différents de temps :

Figure 3.35. Simulation de gravure d'une poutre avec 2 pas différents de temps.

La deuxième étape dans le développement du simulateur était de l'intégrer sous un environnement d'IC, l'éditeur de *Layout* de *Mentor Graphics*. La Figure 3.36 montre une comparaison entre la simulation de gravure anisotrope géométrique réalisée avec le simulateur 2D ACESIM développé et une photo MEB d'une structure à membrane identique dont la gravure est incomplète. On remarque la fidélité du profil simulé.

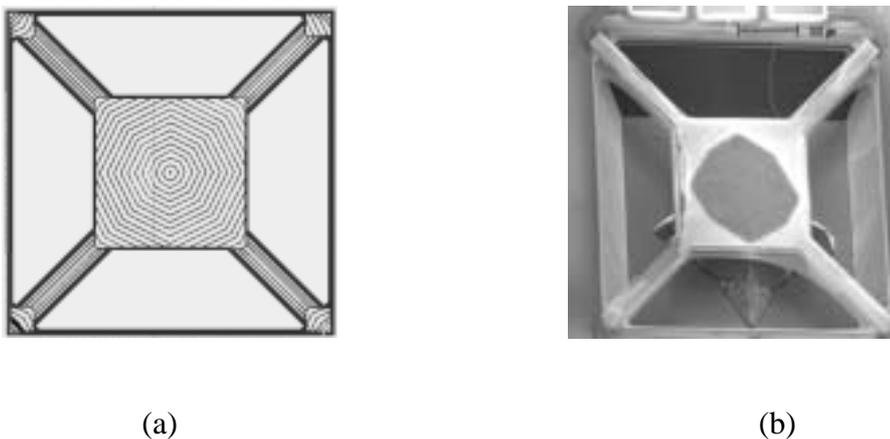


Figure 3.36. Simulation géométrique (a) de la gravure anisotrope d'une membrane soutenue par quatre bras à 45°, et photo MEB (b) de la même structure avec gravure TMAH incomplète.

Il faut noter que nous avons pris en compte que le simulateur de gravure soit capable de produire les résultats de simulation dans un fichier spécifique afin que ces résultats soient exploités dans la phase suivante par le *3D Solide Modeler*.

3.2.2 Construction du modèle 3D

Le terme « modélisation 3D » varie selon le but de l'application. On peut utiliser ce terme pour décrire le processus de visualisation 3D d'un composant microsystème dans le but de vérifier les résultats de gravure par exemple. Ce terme peut être également employé pour décrire un processus beaucoup plus complexe comme la génération d'un modèle à éléments finis d'un composant microsystème pour être simulé ensuite par un simulateur à éléments finis comme ANSYS.

Nos premières expériences dans le domaine de modélisation 3D consistaient à implémenter un outil capable de s'intégrer dans le flot de conception de microsystème dans l'environnement *Mentor Graphics*, c'est-à-dire un outil qui génère une présentation 3D d'un composant microsystème à partir des informations de son layout 2D (CIF), les informations fournies par le simulateur de gravure (s'il y a lieu) et des informations technologiques. Dans un premier temps, notre *3D Solide Modeler* générait des représentations 3D de layout pour être visualisé. Dans les améliorations ajoutées à l'outil, un maillage semi automatique a été implémenté.

La génération d'un modèle 3D est essentiellement contrôlée par un « scénario » qui décrit les différentes étapes du processus technologique utilisé. Ce scénario est détaillé dans un fichier où chaque étape est paramétrée par les deux variables suivants :

- **La nature d'opération :** on spécifie s'il s'agit d'une déposition (*conformal*, *snowfall*, ou *fill*) ou d'une gravure (en surface ou en volume). Il faut noter ici que si l'opération courante est une gravure en volume, on fait appel au simulateur de gravure pour générer la géométrie après-gravure.
- **Le type de couche :** on spécifie ici le type de masque utilisé pendant l'opération courante, ceci afin de récupérer les informations 2D correspondantes du fichier de layout.

Pour illustrer la procédure de description d'un processus technologique et l'exploiter dans la construction de modèle 3D, on a pris l'exemple d'un processus technologique composé des quatre étapes de la Figure 3.37. :

1. **Une déposition « Snowfall » de la couche (a) :** Cette déposition est paramétrée par le type de masque utilisé pour cette couche "*masque (a)*", l'épaisseur de la déposition sur les faces verticales w_1 et l'épaisseur sur les faces inclinées w_2
2. **Une déposition « Fill » de la couche (b) :** Cette déposition est paramétrée par le type de masque utilisé pour cette couche "*masque (b)*" et l'épaisseur de la déposition w_3 .
3. **Une déposition de la couche (c) :** Cette déposition est paramétrée par le type de masque utilisé pour cette couche "*masque (c)*" et l'épaisseur de la déposition w_4 .
4. **Une gravure inclusive en surface :** cette gravure est paramétrée par la profondeur de la gravure w_5 et le nom de masque devant être gravé

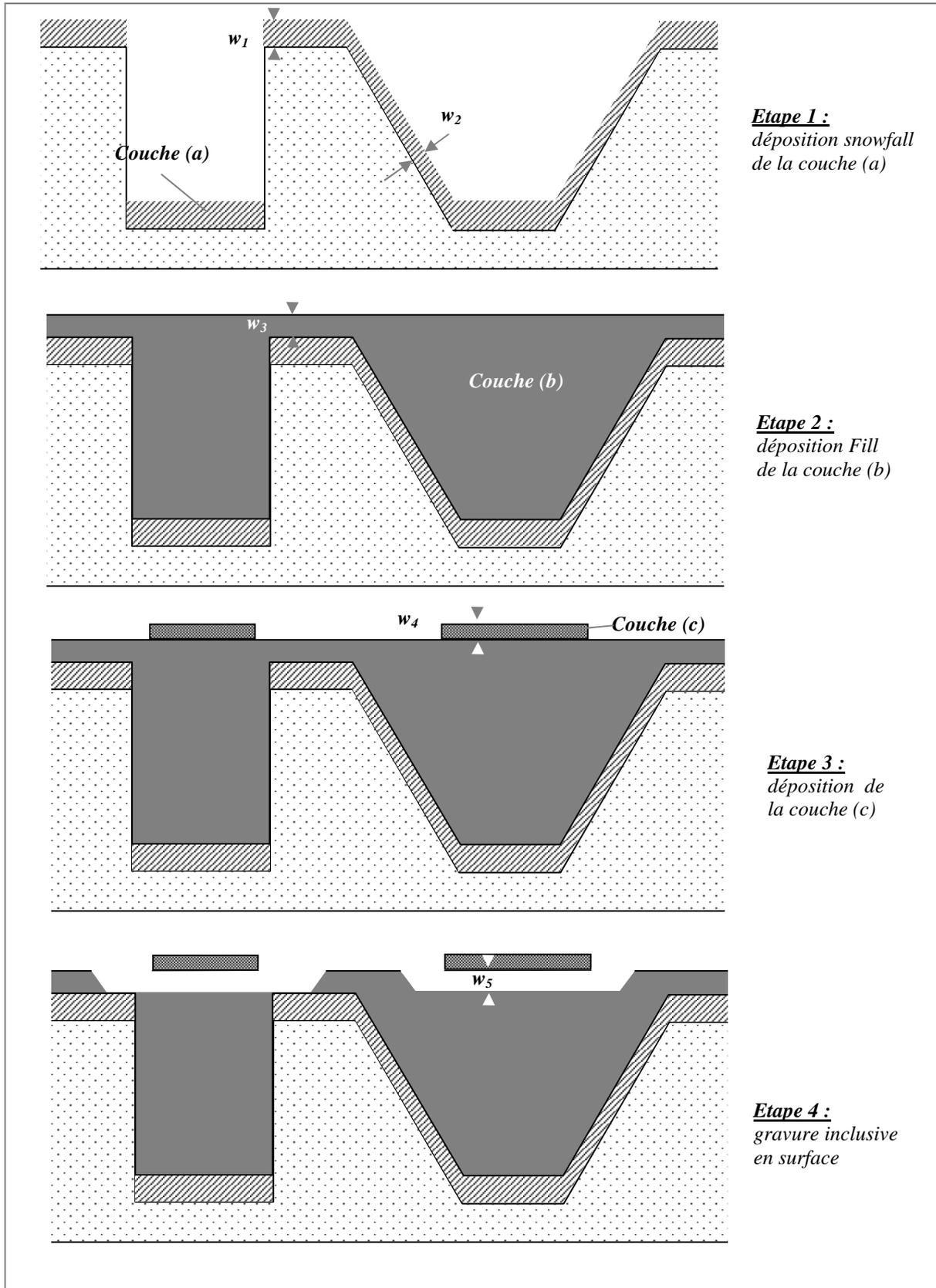


Figure 3.37. Processus technologique de 4 étapes :(1) déposition Snowfall, (2) déposition Fill, (3) déposition du masque de gravure, (4) gravure inclusive en surface.

Un « scénario » qui décrit le processus technologique précédent doit avoir la forme suivante :

```

ProcessName "virtual_process".
Wafer={
  MaskName= "Substrate"
  Thickness= w6
}
Deposit={
  DepositType= SNOWALL
  MaskName= "mask_a"
  ThicknessMax= w1
  ThicknessMin= w2
}
Deposit={
  DepositType= FILL
  MaskName= "mask_b"
  Thickness= w3
}
Deposit={
  DepositType= VOID
  MaskName= "mask_c"
  Thickness= w4
}
Etch={
  EtchType=SURFACE
  MaskName= "mask_c"
  EtchMask= INSIDE
  Depth= w5
  EtchRemove= "mask_b"
}

```

Avec une telle description du processus technologique et si on ajoute les informations 2D fournies pour chaque masque par la description (GDSII, CIF) du layout de composant, on a tout ce qu'il faut pour procéder à une modélisation 3D.

La construction du modèle 3D consiste à combiner la description 2D du composant et la description du procédé. Pour cela il fallait implémenter les deux opérations principales que l'on trouve dans un procédé technologique : la déposition et la gravure. On a adopté un algorithme basé sur la méthode de briques.

La méthode de briques

L'élément de base de cette méthode est la construction d'un modèle 3D d'une épaisseur h à partir d'un polygone quelconque. Ceci pourrait se faire par la construction des faces de la forme 3D correspondante à l'aide d'une extrusion comme suit :

Pour un polygone de N points, chaque deux points successifs p_p, p_{i+p} , construisent une face $[p_i(x_p, y_p, z), p_{i+1}(x_{i+p}, y_{i+p}, z), p'_{i+1}(x_{i+p}, y_p, z+h), p'_i(x_p, y_p, z+h)]$.

Ainsi, la structure 3D aura $N+2$ faces : N faces base de N points + la face du bas et la face du haut (Figure 3.38). La forme résultant de cette opération devient un volume 3D.

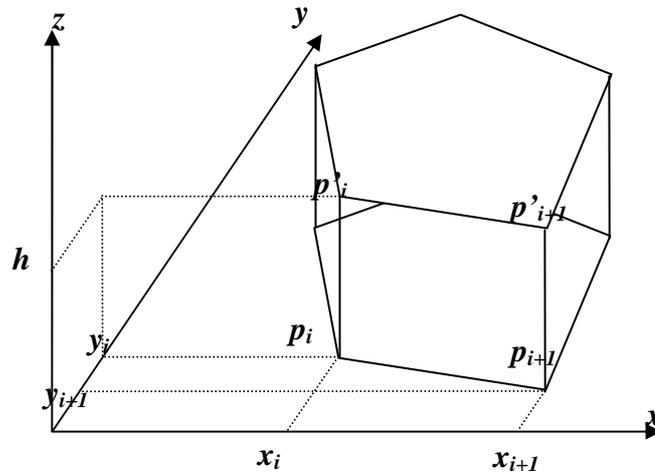


Figure 3.38. Construction d'un volume 3D à partir d'un polygone 2D.

Une opération de déposition pourrait se faire à l'aide de l'algorithme précédent comme suit :

La première couche se fait par extrusion. Un premier volume en résulte. Une deuxième couche pourrait se déposer en utilisant l'intersection et la soustraction entre les formes 2D des bases des deux couches. L'intersection crée des volumes qui se déposeront sur les anciennes couches et d'autres qui seront le résultat de la couche à déposer moins les volumes déjà créés.

La Figure 3.39 montre comment la couche L2 devient l'ensemble des volumes *vol21* et *vol22* dans une déposition *snowfall* où l'intersection des couches (L2, L1) a donné les volumes *vol21* tandis que le volume *vol22* est le volume restant de l'intersection des couches (L1, L2).

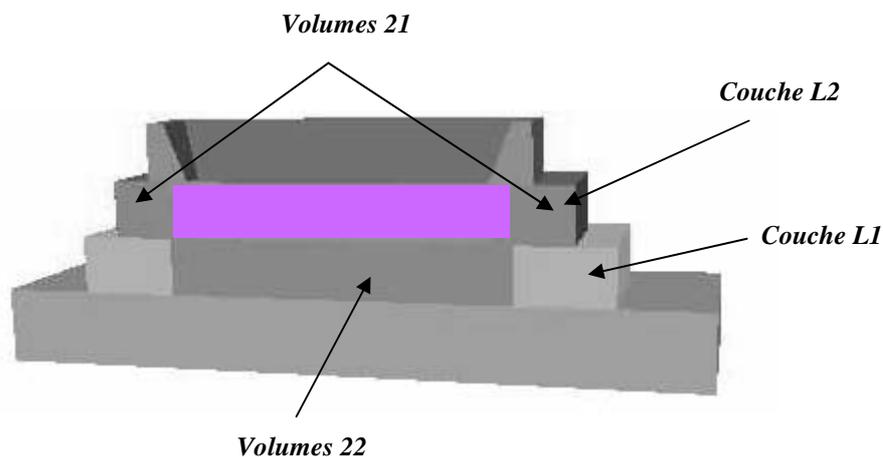


Figure 3.39. Déposition *Snowfall* de couche L2 sur une couche L1.

Une déposition *conformal* de la couche L2 sur la couche L1 va créer les volumes *vol21*, *vol22* et un nouveau volume *vol23* comme le montre la Figure 3.40.

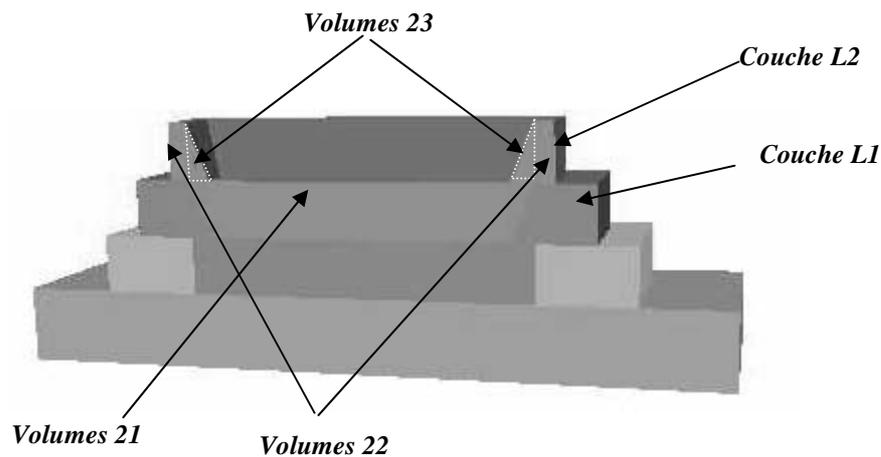


Figure 3.40. Déposition Conforme de couche L2 sur une couche L1.

A noter que la difficulté de cette dernière opération est de calculer la base du volume *vol23*. Ceci se fait en utilisant le facteur *scf*. Il est calculé par l'équation :

$$\text{Base}(\text{vol23}) = \text{base}(\text{vol21}) \cdot \text{scf} - \text{base}(\text{vol21})$$

L'opération d'extension (ou de réduction pour un ratio négatif) d'une forme est une opération cruciale pour le succès de toute simulation de déposition *conforme* ou pour toute gravure non verticale. Souvent les modules géométriques 2D ne disposent pas de tel algorithme (comme Java2D par exemple). Il faut donc l'ajouter, ce qui ne constitue pas un travail évident.

La construction du modèle peut être faite à l'aide d'un module géométrique 2D ou d'un module 3D. Dans les deux cas, le module géométrique doit disposer des opérations géométriques booléennes (intersection, union, etc.).

Les deux solutions (utilisation de module 2D ou de module 3D) ont des avantages et des inconvénients :

- La construction en utilisant des modules géométriques 2D est beaucoup plus rapide puisque les opérations booléennes se font en 2D et il n'y a que la visualisation qui se fait en 3D. Elle souffre de l'inconvénient d'être limitée dans la construction des objets 3D complexes.
- La construction en utilisant des modules géométriques 3D est beaucoup plus simple puisque toutes les opérations se font directement en 3D. Par contre cela rend la construction beaucoup plus lente, un autre inconvénient de cette méthode est d'être contradictoire avec le flot de conception microsysteme puisqu'on part toujours d'un layout 2D.

Finalement les figures suivantes montrent la capacité de *3D modeler* que l'on a implémenté en JAVA2D, différents types de déposition ainsi que les deux types de gravures (en surface et en volume) sont montrés.

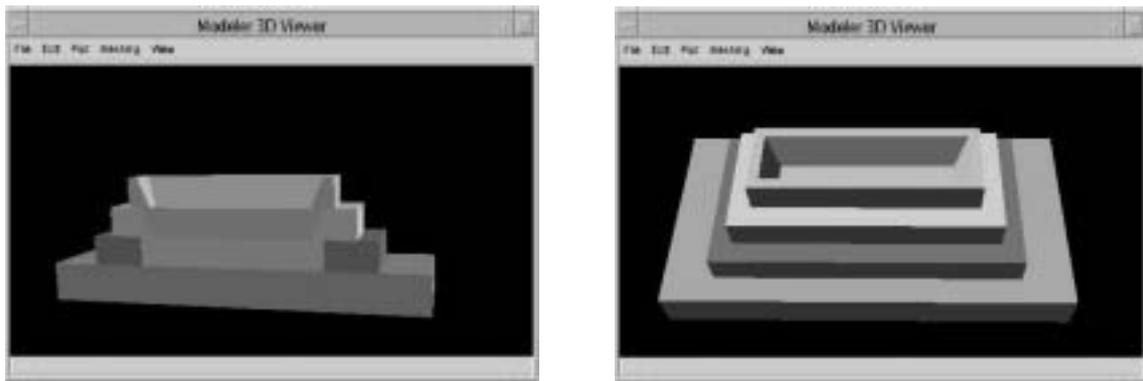


Figure 3.41. *Déposition “ Conformal “ de deux couches et une vue “cross section” de la même structure.*

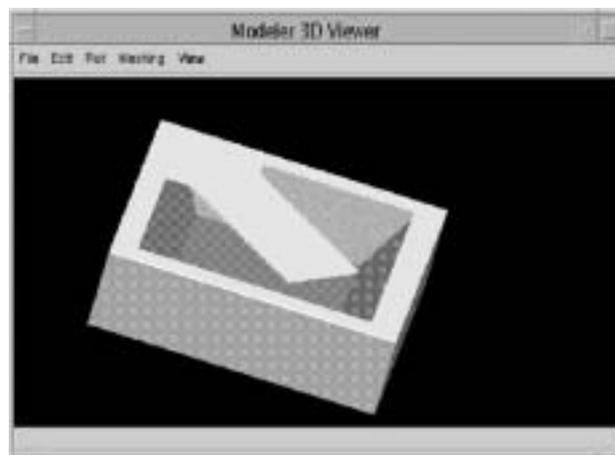


Figure 3.42. *Génération d’une poutre par une gravure anisotrope.*

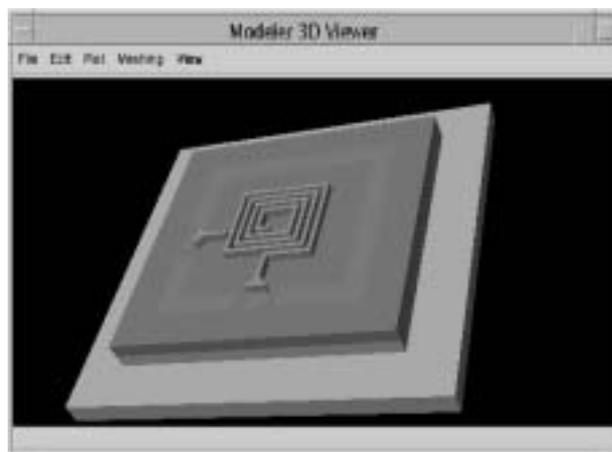


Figure 3.43. *Génération d’une bobine par une gravure en surface.*

3.3 Conclusion :

Dans ce chapitre nous avons donné un aperçu détaillé des différentes technologies de fabrication des microsystemes dans le but d'illustrer l'importance des outils de simulation et de modélisation des procédés technologiques.

Dans le domaine des outils de la modélisation et de la simulation des procédés technologiques, nous nous sommes essentiellement concentrés sur deux outils : le simulateur de gravure anisotrope, et le "*3D Solid Modeler*".

Le simulateur de gravure anisotrope a été développé et intégré d'une manière opérationnelle dans un environnement de CAO de microsysteme.

En ce qui concerne le "*3D Solid Modeler*", nous pensons que des travaux futurs sur cet outil seront nécessaires pour le transformer d'un outil pédagogique en outil professionnel de TCAD.

Parmi les améliorations possibles, nous pensons que l'automatisation du maillage des composants simulés par cet outil est indispensable. Un tel maillage automatique fera le lien entre les outils de la simulation des procédés technologiques et les outils de simulation de microsystemes à base d'éléments finis.

Références du chapitre 3

- [1] O. Tabata, R.A. Sahi, H. Funabashi, K. Shimaoka, S. Sugiyama, "Anisotropic etching of Silicon in TMAH solutions", *In Sensors and actuators A (34)*, pp. 51-57, 1992.
- [2] K.E. Bean, "Anisotropic Etching of Silicon", *In IEEE Trans. Electron Devices*, Vol. ed-25, pp.1185-1192, 1978.
- [3] N. Kaou, J.C. Jeannot, M. de Labachellerie, "New method for reduction of corner undercutting in anisotropic etching, and applications to MEMS and MOEMS", *In DTM99, symposium on design test and microfabrication of MEMS/MOEMS*, Paris, pp.1091-1098, March-April 1999
- [4] D. Veychard, "Conception et réalisation d'un convertisseur électro-thermique à grande constante de temps en technologie microsysteme pour un disjoncteur thermique", Thèse de doctorat, INPG, 1999.
- [5] A. Horn, H. Schröder, E. Obermeier, G. Wachutka, "Simulation of Orientation-Dependent Etching of Silicon Using a New Step Flow Model of 3D Structuring", *In Proc. of 3rd International Conference on Modeling and Simulation of Microsystems, MSM 2000*, San Diego, CA, USA, pp 63-66, 2000.
- [6] http://galaxy.ccsm.uiuc.edu/aces/aces_home.htm
- [7] P. French, M. Nagao, M. Esashi, "Electrochemical etch-stop in TMAH without externally applied bias", *In Sensors and actuators A (56)*, pp 279-280, 1996.
- [8] T.J. Hubbard, E.K. Antonsson, "Etch rate modeling in MEMS design", *In proceedings of the 1996 ASME design engineering technical conference and computer engineering conference*, Irvine, CA, USA, August 18-22, 1996.
- [9] J.M. Paret, "Étude et mise au point de la méthodologie de conception et de fabrication collective de microsystemes sur silicium", *thèse de doctorat, INPG*, Grenoble, France, Janvier 1997.
- [10] CRONOS integrated systems : www.memsrus.com
- [11] SANDIA National labs : www.mdl.sandia.gov
- [12] J.H. Smith, S. Montague, J.J. Sniegowski, J.R. Murray, R.P. Manginell, P.J. McWhorter, "Characterization of the embedded micromechanical device approach to the monolithic integration of MEMS with CMOS", *In SPIE Micromachining and Microfabrication*, 1996.
- [13] W. Yun, R. Howe, P. Gray, "Surface micromachined, digitally force-balanced accelerometer with integrated CMOS detection circuitry", *In Proceeding of the IEEE Solid-State Sensor and Actuator Workshop*, pp. 126, 1992.

- [14] Y. Yee, M. Park, W. Lee, S. Kim, K. Chun, "An integration process of micro electro mechanical polysilicon with CMOS analog/digital circuits", *In Sensors and Actuators A(78)*, pp 120-129, 1999
- [15] Y. B. Gianchandani, H. Kim, M. Shinn, B. Ha, B. Lee, K. Najafi, C. Song, "A MEMS-First Fabrication Process for Integrating CMOS Circuits with Polysilicon Microstructures", *In IEEE MEMS 98 Conference, Heidelberg, January, 27, 1998.*
- [16] J. H. Smith, S. Montague, J. J. Sniegowski, "Material and processing issues for the monolithic integration of microelectronics with surface-micromachined polysilicon sensors and actuators", *In Micromachining and Microfabrication, SPIE, 1995.*
- [17] J. J. Sniegowski, M. P. de Boer, "IC-Compatible Polysilicon Surface Micromachining", *In Annu. Rev. Mater. Sci. 2000, 30, pp 299-333.*
- [18] W. Ehrfeld, F. Götze, D. Münchmeyer, W. Schelb and D. Schmidt, "LIGA Process: Sensor Construction Techniques via x-ray Lithography", *In IEEE Solid-State Sensor and Actuator Workshop Technical Digest*, pp. 1-4, Hilton Head, SC 1988.
- [19] B. Charlot, "Modélisation de fautes et conception en vue du test structurel des microsystemes", Thèse de doctorat, INPG, 2001.
- [20] F.P. Preparata, S.J. Hong, "Convex hull of finite sets of points in two and three dimensions", University of Illinois at Urbana-Champaign, 1977.

4

Modélisation composant et système

4.1 Modélisation par description analytique puis calcul numérique

La représentation du comportement d'un microsysteme est faite, de manière générale, par un système d'équations différentielles à dérivées partielles où les grandeurs physiques dépendent de la position spatiale et du temps. La solution de ce système d'équations demande en général des méthodes de calcul numérique tels que les programmes FEM. Ces méthodes sont très puissantes mais aussi très lourdes à utiliser [6]

Très souvent, il est possible de représenter ces systèmes par des modèles à constantes localisées, prenant le temps comme la seule variable indépendante et donnant lieu à un système d'équations différentielles ordinaires. Ces systèmes peuvent alors être modélisés au moyen de circuits équivalents basés sur l'analogie avec les systèmes électriques. La théorie des réseaux de Kirchhoff décrivant les circuits électriques appliquée sur d'autres domaines peut apporter deux avantages. Premièrement cela permet de profiter des méthodes d'analyse et de calcul déjà disponibles et, deuxièmement, comme les transducteurs comprennent aussi des parties électriques, un même type de représentation facilite considérablement leur étude et leur conception [6].

4.1.1 Constantes localisées réseaux de Kirchhoff

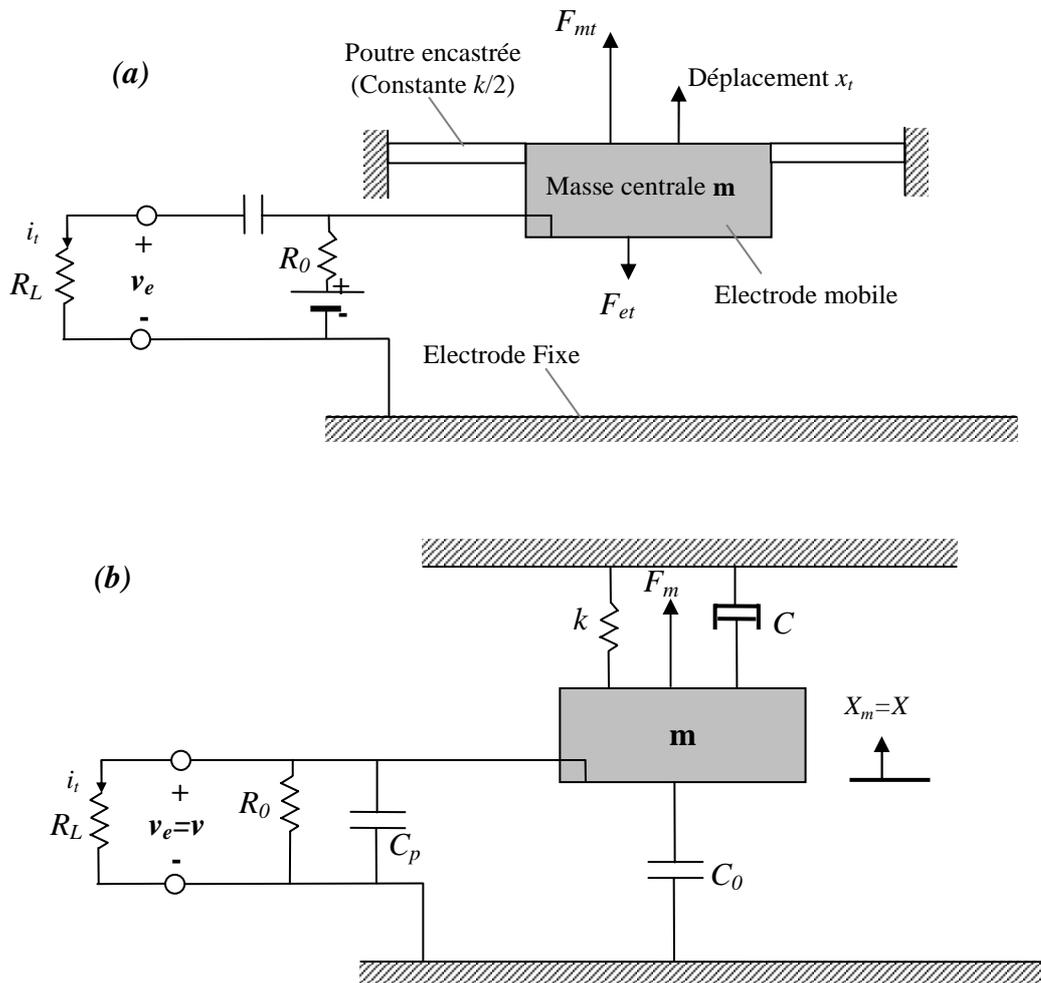
4.1.1.1 Introduction

La caractéristique fondamentale des systèmes à constantes localisées ou à composants discrets (en anglais **lumped-parameter** ou **discrete-parameter model**) est que les propriétés physiques du système telles que la masse, la raideur, l'amortissement, la capacité, l'inductance

sont résumées ou contenues dans des composants discrets ou localisés généralement idéalisés (exemple : une self sans capacité parasite ou résistance série) [6].

Les modèles à constantes localisées sont valables tant que la longueur d'onde des signaux est grande par rapport aux dimensions physiques du dispositif (dans le domaine de l'électricité quand λ devient petite (exemple : $\lambda=10$ cm pour $f=3$ GHz) I et V vérifient l'équation des télégraphistes).

Pour simplifier cette introduction nous nous restreindrons aux systèmes qui ne présentent qu'un seul **degré de liberté** (1 **DDL** et en anglais **SDOF** pour *single degree of freedom*) dans le domaine de la mécanique, ceci se traduit par une seule résonance mécanique. Ce type de modèle est largement utilisé pour différents types de microsystèmes. Notre approche peut être mieux appréhendée par l'exemple de la Figures 4.1.



Figures 4.1. Un transducteur à un seul degré de liberté modélisé par les éléments à constantes localisées.

La structure schématisée est un transducteur de type électrostatique qui peut être utilisé en tant que capteur de force ou encore comme microphone. Le dispositif est constitué de deux poutres encastrées- encastrées (EE et *clamped – clamped* en anglais) avec une masse centrale (le diaphragme du microphone = membrane rigide) libre de se déplacer selon l'axe vertical. Par ailleurs cette membrane mobile constitue aussi l'électrode mobile d'un condensateur plan noté C_0 .

Les poutres encastrées quant à elles, possèdent chacune des constantes de raideur $k/2$ afin que l'ensemble ait une constante de raideur équivalente k .

La membrane se déplace sous la sollicitation d'une force purement mécanique F_{mt} (déplacement x_t) ou bien sous l'action d'une force électrostatique F_{et} qui tend à rapprocher les électrodes entre elles.

Le circuit électrique comporte quant à lui :

- v_0 une tension continue qui polarise le système,
- un condensateur C_d de découplage de la tension continue,
- une résistance R_L qui représente l'impédance d'entrée d'un étage d'amplification

Pour que le système de mesure du signal dynamique issu du transducteur v_e à la pulsation ω ne soit pas perturbé, il faudra choisir R_L et C_d de sorte que $\omega R_L C_d \gg 1$ et $R_L \gg R_0$.

Le schéma de la Figures **4.1 (b)** représente le transducteur modélisé par les éléments à constantes localisées. C_p représente ici une capacité parasite due aux connections électriques nécessaires pour collecter le signal fourni par le transducteur [6].

4.1.1.2 Étude de transducteurs élémentaires

I. Introduction

On transfère de l'énergie au transducteur par des ports (dans ce cas un port pour la partie électrique, un port pour la partie mécanique) représentés par exemple par le quadripôle de la Figure **4.2** [6].



Figure 4.2. Un transducteur électromécanique représenté comme quadripôle

Un port est défini par deux grandeurs une extensive, l'autre intensive. Dans le cas du quadripôle de la Figure **4.2**, nous avons un port associé aux effets mécaniques caractérisés par les variables $u(t) = \dot{x}(t)$ et $F(t)$; et un port électrique caractérisé par $i(t) = \dot{q}(t)$ et $v(t)$. Cette modélisation peut être étendue à plusieurs ports (exemple : modélisation par un hexapôle des transducteurs piézo-électriques), et bien sûr au domaine des transducteurs électromagnétiques et électrodynamiques (variables $i(t)$ et flux magnétique).

II. Configurations de base

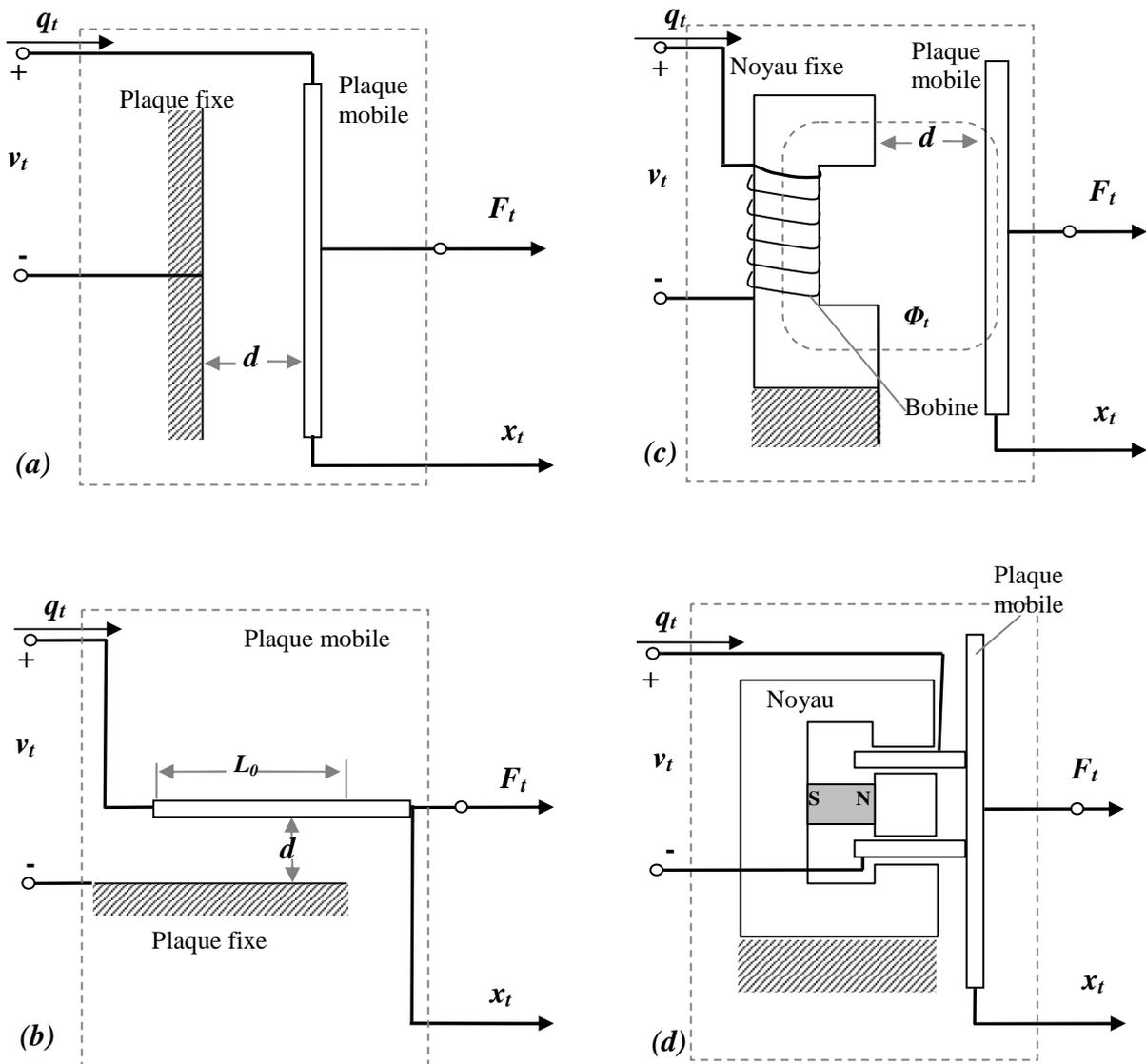


Figure 4.3. Les différentes catégories de transducteurs électromécaniques.

On reporte sur la Figure 4.3, quatre exemples de transducteurs électromécaniques à constantes localisées. On distingue sur les Figures 4.3 (a) et 4.3 (b) les transducteurs électrostatiques avec des déplacements transverses et parallèles des électrodes. Par ailleurs, les transducteurs électromagnétiques sont eux aussi décomposés en deux catégories celle où la bobine de champ est fixe (électromagnétique), et une autre où la bobine est mobile (électrodynamique).

Dans la suite de ce chapitre, nous nous attarderons sur la modélisation du transducteur de la Figure 4.3 (a).

III. Étude de transducteurs électrostatiques

On considère le transducteur de la Figure 4.3 (a) où les variables d'état sont dans ce cas x_t le déplacement et q_t la charge du condensateur variable. Dans notre étude dynamique les variables dépendent de t , mais pour alléger les écritures nous omettrons l'écriture de cette variable [6].

L'énergie de nature électrostatique emmagasinée par le transducteur s'écrit :

$$(eq.1) \quad W_e = W_e(q_t, x_t) = \frac{q_t^2}{2 C(x_t)} = \frac{q_t^2 (d + x_t)}{2 \varepsilon_0 A_e}$$

où $C(x_t) = \frac{\varepsilon_0 A_e}{d + x_t}$ est le condensateur variable qui dépend de x_t . A_e est la surface des

électrodes en regard, d est l'entrefer du condensateur non chargé.

La différentielle totale de W_e donne :

$$(eq.2) \quad dW_e = \left(\frac{\delta W_e}{\delta q_t} \right) dq_t + \left(\frac{\delta W_e}{\delta x_t} \right) dx_t$$

Pour un transducteur à l'équilibre thermodynamique l'énergie amenée par les deux ports est donnée par :

$$(eq.3) \quad dW_e = v_t dq_t + F_t dx_t$$

où v_t est la tension entre les électrodes et F_t est la force mécanique agissant sur l'électrode mobile. En soustrayant les relations (2) et (3), nous obtenons une relation qui est satisfaite lorsque les coefficients de dq_t et dx_t sont nuls d'où :

$$(eq.4) \quad v_t(q_t, x_t) = \left(\frac{\delta W_e}{\delta q_t} \right)_{x_t=cte} = \frac{q_t (d + x_t)}{\varepsilon_0 A_e}$$

$$(eq.5) \quad F_t(q_t, x_t) = \left(\frac{\delta W_e}{\delta x_t} \right)_{q_t=cte} = \frac{q_t^2}{2 \varepsilon_0 A_e}$$

F_t est la force extérieure appliquée sur l'électrode mobile pour vaincre la force électrostatique (attractive).

F_t varie de façon quadratique avec la variable q_t , ce qui entraîne un comportement **non linéaire du système**.

La **linéarisation** du système est obtenue en introduisant une tension continue (**DC bias**) ou bien par des **conditions d'équilibre** grâce à des forces extérieures. Les équations qui décrivent les relations linéaires entre des accroissements (ou petits signaux) des variables intensives par rapport aux variables d'état (ici x et q) autour d'un point de polarisation défini par x_0 et q_0 , sont appelées **équations constitutives** du système.

$$(eq.6) \quad v(q, x) = \left(\frac{\delta v_t}{\delta q_t} \right)_0 q + \left(\frac{\delta v_t}{\delta x_t} \right)_0 x = \frac{(d + x_0)}{\varepsilon_0 A_e} q + \frac{q_0}{\varepsilon_0 A_e} x = \frac{1}{C_0} q + \frac{v_0}{x_0} x$$

$$(eq.7) \quad F(q, x) = \left(\frac{\delta F_t}{\delta q_t} \right)_0 q + \left(\frac{\delta F_t}{\delta x_t} \right)_0 x = \frac{q_0}{\varepsilon_0 A_e} q + 0x = \frac{v_0}{x_0} q + 0x$$

où

$$q_0 = C_0 v_0 = \frac{\varepsilon_0 A_e v_0}{x_0}$$

C_0 est la capacité statique du transducteur et v_0 la polarisation continue appliquée.

Le modèle de ce transducteur n'est pas satisfaisant dans la mesure où il est instable. Cet inconvénient peut être résolu si on tient compte de la présence d'une force de rappel sur l'électrode mobile (matérialisée ici par un ressort de constante de raideur k).

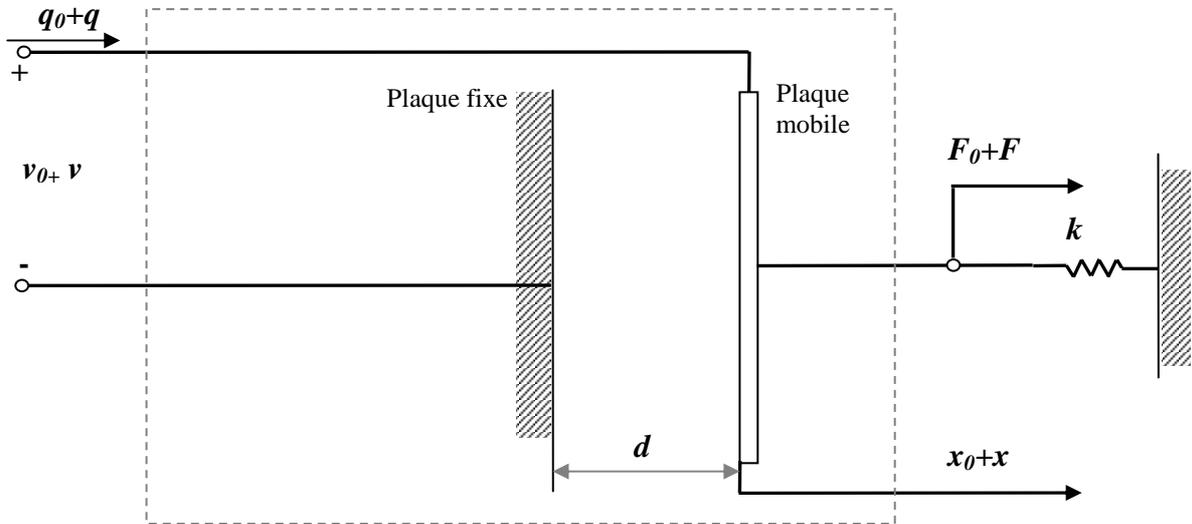


Figure 4.4. Un modèle modifié d'un transducteur électromécanique pour résoudre le problème d'instabilité.

La présence de ce ressort se traduit par l'introduction d'un terme supplémentaire dans l'équation (1) :

$$(eq.8) \quad W_{em} = W_{em}(q_t, x_t) = \frac{q_t^2}{2C(x_t)} + \frac{1}{2}k(x_t - x_r)^2 = \frac{q_t^2(d + x_t)}{2\varepsilon_0 A_e} + \frac{1}{2}k(x_t - x_r)^2$$

où x_r est associé à la position au repos du ressort.

L'équation (4) n'est pas affectée, mais l'équation (5) devient :

$$(eq.9) \quad F_t(q_t, x_t) = \frac{q_t^2}{2\varepsilon_0 A_e} + k x_t$$

De même l'équation (7) devient :

$$(eq.10) \quad F(q, x) = \left(\frac{\delta F_t}{\delta q_t} \right)_0 q + \left(\frac{\delta F_t}{\delta x_t} \right)_0 x = \frac{q_0}{\varepsilon_0 A_e} q + k x = \frac{v_0}{x_0} q + k x$$

On peut aussi choisir comme variables la tension v et le déplacement x , on alors les équations suivantes:

$$(eq.11) \quad q(v, x) = \frac{\varepsilon_0 A_e}{(d + x_0)} v - \frac{q_0}{(d + x_0)} x = \frac{\varepsilon_0 A_e}{(d + x_0)} v - \frac{v_0 \varepsilon_0 A_e}{(d + x_0)^2} x$$

$$(eq.12) \quad F(v, x) = \frac{q_0}{(d + x_0)} v + \left(k - \frac{q_0^2}{\varepsilon_0 A_e (d + x_0)} \right) x = \frac{v_0 \varepsilon_0 A_e}{(d + x_0)^2} v + \left(k - \frac{v_0^2 \varepsilon_0 A_e}{(d + x_0)^3} \right) x$$

L'équation (12) permet de dire que le système sera stable tant que $k > k'$ où $k' = \frac{v_0^2 \varepsilon_0 A_e}{(d + x_0)^3}$

La stabilité d'un tel transducteur dépend donc de v_0^2 . On considèrera dans la suite des

signaux sinusoïdaux de pulsation ω et on utilisera la notation complexe (exemple : $x(t)=x \cdot \exp(j\omega t)$). Tout autre signal pouvant être étudié à l'aide de la transformée de Fourier.

IV. Équations caractéristiques

Les équations caractéristiques d'un transducteur électromécanique décrivent les relations linéaires qui existent en régime de petits signaux autour d'un point de polarisation. Ces équations sont convenablement représentées par l'algèbre des matrices [6].

Pour le transducteur qui nous intéresse on note que :

- $q(t)$, $x(t)$ sont les variables d'états,
- $v(t)$, $F(t)$ sont les variables intensives,
- $i(t) = \dot{q}(t)$ et $u(t) = \dot{x}(t)$ sont les variables extensives.

On peut écrire, comme pour un quadripôle électrique classique, six relations avec cette configuration.

A partir des relations (11) et (12), nous pouvons obtenir la matrice [B] liant $v(t)$ et $F(t)$ à $q(t)$ et $x(t)$.

Ces équations deviennent :

$$(eq.13) \quad v(q, x) = \frac{(d + x_0)}{\epsilon_0 A_e} q + \frac{v_0 \epsilon_0 A_e}{(d + x_0)} x = \frac{1}{C_0} q + \frac{\Gamma}{C_0} x$$

et

$$(eq.14) \quad F(q, x) = \frac{\Gamma}{C_0} q + k x$$

$C_0 = \frac{\epsilon_0 A_e}{d + x_0} = \frac{q_0}{v_0}$ (Farad) est la capacité statique du condensateur et

$\Gamma = \frac{v_0 \epsilon_0 A_e}{(d + x_0)^2}$ ($N V^{-1}$) est le facteur de transduction.

On déduit aussi le coefficient de couplage statique $k (\omega \rightarrow 0) k = \sqrt{\frac{\Gamma^2}{C_0}}$

L'expression de [B] devient alors $[B] = \begin{bmatrix} \frac{1}{C_0} & \frac{\Gamma}{C_0} \\ \frac{\Gamma}{C_0} & k \end{bmatrix}$

Pour la matrice de transfert [T] nous avons $[T] = \begin{bmatrix} \frac{1}{\Gamma} & \frac{-1}{j\omega \Gamma} \left(k - \frac{\Gamma^2}{C_0} \right) \\ \frac{j\omega C_0}{\Gamma} & \frac{-k C_0}{\Gamma} \end{bmatrix}$

De même pour les transducteurs de la Figure 4.3 (b,c,d), il est possible d'écrire les matrices [B] et [T] qui sont reportées dans le Tableau 4.1. Le Tableau 4.2 fournit les expressions des coefficients de couplage et de transduction pour les quatre configurations possibles de transducteurs.

Type de transducteur	Etat q_1, q_2	Efforts θ_1, θ_2	Flots f_1, f_2	$\begin{Bmatrix} \theta_1 \\ \theta_2 \end{Bmatrix} = B \begin{Bmatrix} q_1 \\ q_2 \end{Bmatrix}$	$\begin{Bmatrix} \theta_1 \\ f_1 \end{Bmatrix} = T \begin{Bmatrix} \theta_2 \\ f_2 \end{Bmatrix}$
Electrostatique (transverse, Figure 4.3 (a))	$q(t), x(t)$	$v(t), F(t)$	$i(t) = \dot{q}(t), u(t)$	$\begin{bmatrix} \frac{1}{C_0} & \frac{\Gamma}{C_0} \\ \frac{\Gamma}{C_0} & k \end{bmatrix}$	$\begin{bmatrix} \frac{1}{\Gamma} & \frac{-1}{i\omega\Gamma} (k - \frac{\Gamma^2}{C_0}) \\ \frac{i\omega C_0}{\Gamma} & -\frac{kC_0}{\Gamma} \end{bmatrix}$
Electrostatique (parallèle, Figure 4.3 (b))	$q(t), x(t)$	$v(t), F(t)$	$i(t) = \dot{q}(t), u(t)$	$\begin{bmatrix} \frac{1}{C_0} & \frac{\Gamma}{C_0} \\ \frac{\Gamma}{C_0} & k + \frac{\Gamma^2}{C_0} \end{bmatrix}$	$\begin{bmatrix} \frac{1}{\Gamma} & \frac{-k}{i\omega\Gamma} \\ \frac{i\omega C_0}{\Gamma} & -\frac{k}{\Gamma} (C_0 + \frac{\Gamma^2}{k}) \end{bmatrix}$
Electromagnétique (Figure 4.3 (c))	$\lambda(t), x(t)$	$i(t), F(t)$	$\dot{\lambda}(t) = v(t), u(t)$	$\begin{bmatrix} \frac{1}{L_0} & \frac{\Psi}{L_0} \\ \frac{\Psi}{L_0} & k \end{bmatrix}$	$\begin{bmatrix} \frac{1}{\Psi} & \frac{-1}{i\omega\Psi} (k - \frac{\Psi^2}{L_0}) \\ \frac{i\omega L_0}{\Psi} & -\frac{kL_0}{\Psi} \end{bmatrix}$
Electrodynamique (Figure 4.3 (d))	$\lambda(t), x(t)$	$i(t), F(t)$	$\dot{\lambda}(t) = v(t), u(t)$	$\begin{bmatrix} \frac{1}{L_0} & \frac{\Psi}{L_0} \\ \frac{\Psi}{L_0} & k + \frac{\Psi^2}{L_0} \end{bmatrix}$	$\begin{bmatrix} \frac{1}{\Psi} & \frac{-k}{i\omega\Psi} \\ \frac{i\omega L_0}{\Psi} & -\frac{k}{\Psi} (L_0 + \frac{\Psi^2}{k}) \end{bmatrix}$

Tableau 4.1. Les matrices $[B]$ et $[T]$ pour les quatre configurations possibles de transducteurs

Type de transducteur	Composants statiques	Facteurs de transduction $\Gamma [NV^{-1} = A(ms^{-1})^{-1}]$ et $\Psi [NA^{-1} = V(ms^{-1})^{-1}]$	Statique ($\omega \rightarrow 0$) Facteur de couplage kf -]
Electrostatique (transverse, Figure 4.3 (a))	$C_0 \equiv \frac{q_0}{v_0} = \frac{\epsilon_0 A_e}{d + x_0}$	$\Gamma = \frac{q_0}{d + x_0} = \frac{\epsilon_0 A_e v_0}{(d + x_0)^2}$	$\sqrt{\frac{\Gamma^2}{C_0 k}}$
Electrostatique (parallèle, Figure 4.3 (b))	$C \equiv \frac{q_0}{v_0} = \frac{\epsilon_0 (l_0 - x_0) h}{d}$	$\Gamma = \frac{q_0}{l_0 - x_0} = \frac{\epsilon_0 h v_0}{d}$	$\sqrt{\frac{1}{1 + \frac{C_0 k}{\Gamma^2}}}$
Electromagnétique (Figure 4.3 (c))	$L_0 \equiv \frac{\lambda_0}{i_0} = \frac{N^2 \mu_0 A_e}{d + x_0}$	$\Psi = \frac{\lambda_0}{d + x_0} = \frac{N^2 \mu_0 A_e i_0}{(d + x_0)^2}$	$\sqrt{\frac{\Psi^2}{L_0 k}}$
Electrodynamique (Figure 4.3 (d))	$L_0 = L_s = \text{inductance des bobines en série}$	$\Psi = \beta_0 l$	$\sqrt{\frac{1}{1 + \frac{L_0 k}{\Psi^2}}}$

Tableau 4.2. Les coefficients couplage et de transduction pour les quatre configurations possibles de transducteurs

Le coefficient de couplage κ est une grandeur fondamentale qui reflète la capacité de conversion énergétique (électrique/mécanique) du transducteur sans pertes. $\kappa = 0$ correspond à une interaction nulle entre les deux types d'énergie. κ varie théoriquement entre 0 et 1, et du point de vue expérimental κ est généralement compris entre 0,05 et 0,25 [6].

4.2 Modélisation par éléments finis

L'analyse en éléments finis ou FEM est un outil mathématique qui a été développé à l'origine pour le calcul de structures mécaniques complexes et pour en évaluer la résistance à la rupture. Ces outils sont aujourd'hui très utilisés dans le domaine de la mécanique, de l'aérospatial, du génie civil, de l'automobile, etc. Il est possible grâce à ces méthodes numériques d'effectuer des analyses de structures en mode statique ou dynamique dans les domaines linéaires et non linéaires mais également des analyses thermiques, de mécanique des fluides et d'électromagnétisme. La nature multidisciplinaire des microsystèmes fait que ces outils d'analyse sont très utilisés pour la conception des microstructures.

Le principe de l'analyse en éléments finis est de rechercher une solution approchée de la solution exacte de la distribution d'une grandeur physique sur un domaine donné. Pour cela le domaine est divisé en sous domaines que l'on appelle éléments. On appelle cette opération la discrétisation. Cela revient à diviser un problème complexe en une quantité de petits problèmes simples à résoudre [3].

Les éléments sont des ensembles géométriques de points appelés nœuds et qui contiennent une représentation mathématique des échanges d'énergie et une fonction d'interpolation du champ à l'intérieur de l'élément et entre ses nœuds. Le champ global calculé est alors déterminé par un nombre fini de valeurs du champ sur les nœuds. Les inconnues en chacun des nœuds sont appelées variables nodales ou degrés de liberté.

Une analyse en éléments finis se décompose en plusieurs étapes :

- Tout d'abord le choix du domaine d'énergie permet de fixer la nature des variables que l'on appelle les degrés de liberté. Ce sont par exemple les déplacements aux nœuds dans l'analyse structurelle, la température dans l'analyse thermique ou le potentiel dans l'analyse électrostatique.
- La description du domaine géométrique permet ensuite la discrétisation de ce dernier en éléments et en nœuds. Cette opération s'appelle le maillage [1] [3]
- La préparation du système d'équations global à résoudre se fait en assemblant les matrices et les vecteurs associés à chaque élément, puis en incorporant les conditions aux limites et les charges dans le système d'équations global. Cette étape précède la résolution du système d'équations qui a pour but d'extraire les valeurs des degrés de liberté à chaque nœud [4].
- La dernière étape est le calcul des grandeurs associées aux degrés de liberté comme les contraintes, les courants, les flux de chaleur, etc.

Les éléments dans l'analyse en éléments finis sont les briques élémentaires dont le maillage va représenter le système géométrique à simuler. Les éléments sont donc des primitives géométriques comportant plusieurs nœuds. La Figure 4.5 représente des exemples d'éléments 2D ou 3D.

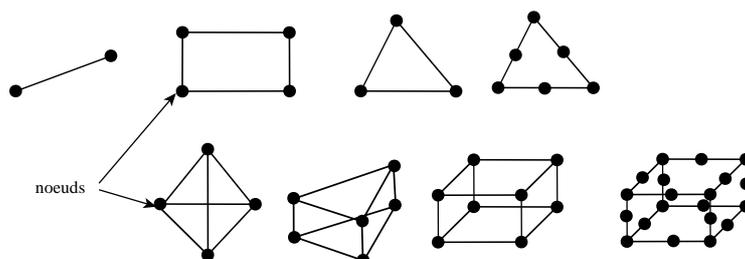


Figure 4.5. Exemples d'éléments assemblés en primitives géométriques.

Le maillage consiste à discrétiser un domaine géométrique en un ensemble d'éléments géométriques primitifs. La Figure 4.6 montre différentes solutions de maillage appliquées à une structure à 3 dimensions. La Figure 4.6 (a) représente un maillage avec des éléments cubiques alors que le maillage des Figure 4.6 (b,c,d) a été réalisé avec des tétraèdres. On note qu'il est possible d'ajuster la densité d'éléments de manière uniforme (Figure 4.6 (b,d)) et non uniforme (Figure 4.6 (c)) de manière à augmenter la précision des calculs dans des zones sensibles. Le problème le plus fréquent dans le maillage géométrique est la concomitance des nœuds qui doit assurer la continuité du matériau [2] [3].

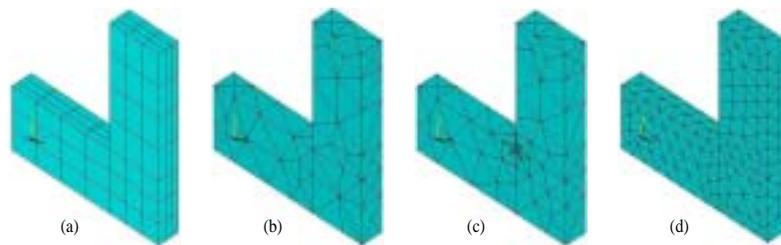


Figure 4.6. Différents types de maillage : avec éléments cubiques (a), avec éléments tétraédriques (b), avec maillage adaptatif (c) et avec maillage plus fin (d).

4.3 La simulation comportementale des microsystèmes

Les simulateurs électriques analogiques tels que SPICE permettent de simuler des circuits contenant différents éléments tels que des résistances, des transistors, des diodes, des capacités et des inductances. Les modèles de ces composants sont le plus fréquemment inclus dans le code du simulateur et la création du circuit se fait en composant une *netlist* qui va décrire le réseau formé par tous les éléments [5].

4.3.1 Simulateurs de circuits

Pour simuler le circuit électrique, le simulateur va former son système d'équations à partir des lois de Kirchoff en introduisant les variables qui sont les potentiels aux nœuds v_k et les courants de branche i_k l'équation globale peut alors se ramener sous forme matricielle :

$$(eq.15) \quad [Y]\{v\} = \{I\}$$

$\{I\}$ étant le vecteur contenant toutes les sources de courant indépendantes et $[Y]$ la matrice admittance.

Ce système d'équations doit être modifié si le circuit contient des sources de tension ou de courant contrôlées. A partir de ce système d'équations il existe plusieurs modes de calcul dépendant du type d'analyse souhaité, on remarque :

- L'analyse statique (DC) ou point de fonctionnement, qui est la résolution d'un système d'équations différentielles. Dans la majorité des cas la présence d'éléments non linéaires impose l'emploi des algorithmes qui vont linéariser les équations autour d'un point d'équilibre de manière itérative, c'est par exemple l'algorithme de Newton-Rhapson ;
- L'analyse harmonique (AC), se traite par la représentation des grandeurs en complexes linéarisées autour d'un point de fonctionnement. Il est également possible d'extraire les valeurs propres du circuit par une analyse des pôles et des zéros ;

- L'analyse transitoire (transient) qui permet d'analyser le circuit au cours du temps se fait en discrétisant l'intervalle de temps en un nombre fini de points et en résolvant le système d'équations non linéaires à chaque pas de temps. Le pas de temps est contrôlé dynamiquement pour s'adapter à la dynamique des variables.
- Enfin il est possible d'effectuer des analyses statistiques suivant des dispersions de paramètres de composants avec les analyses Monte-Carlo.

La simulation mixte permet de simuler en même temps des circuits analogiques (simulation temporelle) et numériques (simulation événementielle). Les tâches spécifiques de ces simulateurs sont le partitionnement qui est la répartition des blocs analogiques et numériques et la synchronisation des algorithmes de résolution analogiques et numériques qui gèrent le temps différemment.

Les simulateurs électriques analogiques sont basés sur les lois de Kirchoff qui sont elles-mêmes basées sur la loi de conservation de l'énergie qui peut également s'appliquer à d'autres domaines tels que la thermique, la mécanique, la pneumatique, etc. Les tensions et les courants d'un circuit électrique sont l'application au domaine d'énergie électrique d'un système de variables de type potentiel/flux que l'on définit comme suit :

- Une variable de potentiel (*across*) définit une grandeur intrinsèque ou ses dérivées par rapport au temps, par rapport à une référence qui est propre à chaque nœud du réseau ;
- Une variable de flux (*through*) définit une grandeur absolue et directionnelle entre deux nœuds.

4.3.2 La modélisation par langage comportemental

La modélisation comportementale des microsystèmes se base sur l'utilisation des langages de description du matériel. Ces langages permettent de modéliser des circuits intégrés en vue de la validation par simulation. Les langages de description du matériel analogiques comme HDL-A, MAST et VERILOG-A sont destinés à créer des modèles fonctionnels de haut niveau d'abstraction en vue de la simulation. La simulation de circuits complexes au niveau des composants élémentaires (transistors) pose des problèmes de convergence et de temps de calcul. Le Tableau 4.3 résume les différents niveaux de hiérarchie que l'on trouve dans le domaine de la simulation analogique [5].

Niveau d'abstraction	Primitives de simulation	Unités de Conception
Système	Fonctions de transfert Domaine fréquentiel Domaine temporel	Architecture Topologie Connectivité
Fonctionnel	Equations algébriques linéaires et non-linéaires Courbes de transfert Tables	Sommateur, Intégrateur, Multiplieur, Bistable, Valeur absolue
Circuit	Macro-modèles: sources contrôlées éléments passifs dispositifs non-linéaires	Amplis Opérationnels, Sources, Comparateurs
Composant	MOS, BJT, éléments passifs	Propriétés physiques ou géométriques, Tension, Courant

Tableau 4.3. Niveaux hiérarchiques de la simulation analogique

Les langages de description du matériel permettent de créer des modèles de composants en définissant des relations mathématiques entre les variables (potentiel et flux) de leurs entrées sorties. La plupart de ces langages permettent également de définir un domaine d'énergie pour les variables, ils sont donc tout indiqués pour la modélisation des microsystèmes en vue de la simulation avec la partie électronique. De nouveaux standards comme les langages VHDL-AMS et VERILOG-AMS sont amenés à unifier les descriptions numériques, analogiques et microsystèmes [5].

4.3.2.1 Structure d'un modèle HDL

Un modèle comportemental se compose toujours de deux parties :

- Le bloc entité ou module qui décrit les relations entre le modèle et l'extérieur, c'est à dire le nombre et la nature des entrées/sorties (les plots). Dans cette partie il faut déclarer le nombre et la nature des entrées/sorties, les *generics* qui sont les paramètres utilisés dans le modèle et accessibles depuis le simulateur et les *couplings* qui sont les variables qui peuvent être partagées entre plusieurs modèles ;
- Le bloc architecture décrit le comportement du système défini par le bloc entité. Les relations décrites dans ce bloc permettent de lire et d'écrire les grandeurs définies par le bloc entité et de faire des opérations mathématiques soit directement entre ces grandeurs soit par l'intermédiaire de variables locales.

Dans la modélisation par HDL il est possible d'affecter une nature à une broche d'entrée/sortie, c'est à dire de désigner le domaine d'énergie qui définit les unités des variables flux et potentiel. Il est donc possible de déclarer une broche comme étant de nature mécanique (force/déplacement ou force/vitesse) ou de nature thermique (flux de chaleur/température). Dans Verilog-A il est même possible de créer une nature en définissant les variables flux et potentiel.

Dans les langages HDL il est possible d'effectuer deux types d'assignements :

- Les assignements explicites qui affectent une variable connue à une variable inconnue, par exemple : lecture et écriture d'une quantité flux, ou potentiel d'une broche, ou affectation d'une variable locale à une valeur ;
- Les assignements implicites qui nécessitent la résolution d'une équation ou d'un système d'équations.

Remarque, il n'est pas possible d'écrire ou de lire à la fois une variable flux ou potentiel.

Les langages HDL comportent en général un ensemble d'opérateurs mathématiques, comme la dérivation ou l'intégration, ainsi qu'une bibliothèque de fonctions mathématiques.

4.3.2.2 Exemple électrothermique convertisseur électrothermique

Nous allons montrer ici comment il est possible de simuler un microsystème avec un simulateur électrique en utilisant les équivalences électrique / thermique. Le convertisseur électrothermique est un système contenant une résistance chauffante au bout d'une poutre suspendue elle-même contenant une thermopile, reliant l'extrémité de la poutre au substrat. La Figure 4.7 montre une image MEB de convertisseur électrothermique fabriqué en technologie de micro-usinage de volume à partir du procédé CMOS 0,6 μ m d'AMS. La poutre fait environ 250 μ m de long sur 100 μ m de large et comporte en ensemble de 16 thermocouples poly/alu mesurant la différence de température entre la résistance chauffante et le substrat.

• **Analogie électrique/thermique**

Au point de vue comportemental, le système est un quadripôle électrique ayant le comportement d'un système du premier ordre avec une constante de temps définie par le produit de la résistance thermique de la poutre et de sa capacité thermique. Les applications d'un tel système vont de la conversion AC/DC à l'isolation galvanique vers des systèmes disjoncteurs thermiques.

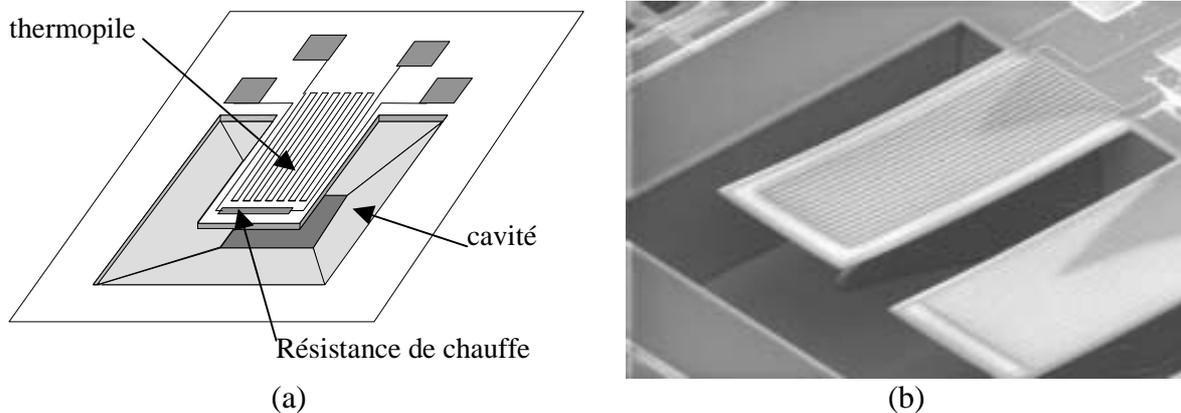


Figure 4.7. Schéma (a) et image MEB (b) d'un convertisseur électrothermique à poutre suspendue.

La modélisation en équivalent électrique d'un tel système est représentée sur la Figure 4.8. Le modèle est scindé en trois parties qui modélisent les parties électriques et la partie thermique. Les parties électriques représentent les résistances électriques, la résistance de chauffe et la résistance interne de la thermopile. La tension Seebeck est modélisée par un générateur de tension contrôlé en tension par l'équivalent électrique de la température de la jonction chaude et multipliée par la différence de coefficients Seebeck. La partie d'échange thermique est modélisée comme une ligne de transmission de la chaleur à une dimension. La tension représente la température et le courant le flux de chaleur. La poutre est discrétisée selon sa longueur en un nombre fini d'éléments thermiques représentant les pertes par conduction (R_{cond}), par convection (R_{conv}) ainsi que la chaleur emmagasinée (C_{th}). La puissance thermique injectée par la résistance est modélisée par un générateur de courant contrôlé en tension ($P=V^2/R$) [17].

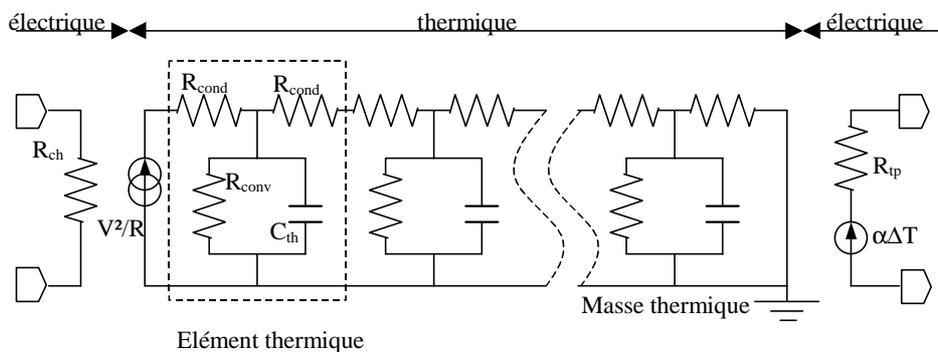


Figure 4.8. Modélisation en équivalent électrique d'un convertisseur électrothermique.

La modélisation par équivalent électrique permet de simuler des systèmes fonctionnant dans des domaines d'énergies différents en utilisant un simulateur électrique. Cependant cette manière de modéliser a des limites et par exemple il n'est pas possible dans l'exemple

précédent de modéliser les pertes par radiation (qui dépendent de ΔT^4). L'utilisation de langages de description du matériel offre plus de possibilités de modélisation [17].

4.3.2.3 Exemple électromécanique : les micro-résonateurs électrostatiques en peigne

Nous allons décrire ici en détail un composant microsysteme, le micro-résonateur électrostatique à électrodes en peigne. Ce composant, usuellement fabriqué en technologie de micro-usinage en surface, est très fréquemment utilisé dans des systèmes plus complexes pour ses propriétés de transduction électrique mécanique comme moteur linéaire. Une autre application est le filtrage électrique en utilisant ce composant comme filtre électromécanique. Le principal intérêt de ce composant est le très grand coefficient de qualité atteint lors de la résonance.

• Description du composant

Les micro-résonateurs en peigne sont des systèmes électromécaniques composés de deux électrodes constituées de peignes interdigités et d'une partie mobile pouvant se déplacer sur un axe. La Figure 4.9 montre une représentation schématique du composant. La masse mobile suspendue au-dessus du substrat et reliée à ce dernier par l'intermédiaire de ressorts de flexion est le cœur du système oscillant. Les deux électrodes en peigne effectuent la transduction électrique mécanique et inversement. Le résonateur harmonique constitué de la masse mobile et des ressorts effectue le filtrage [18] [19].

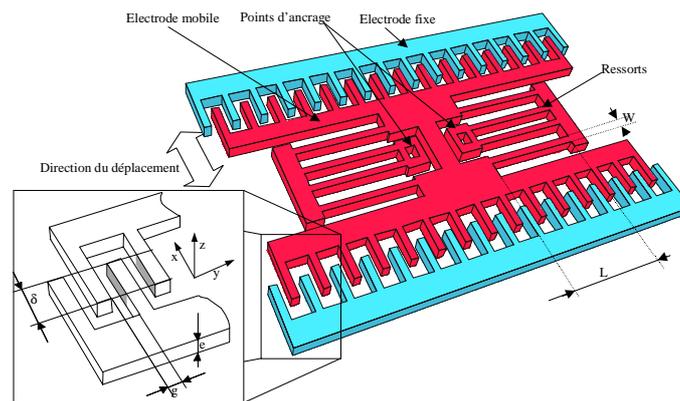


Figure 4.9. Schéma de principe du fonctionnement d'un micro-résonateur en peigne.

Au niveau électrique le micro-résonateur est constitué de trois pôles, deux électrodes fixes et l'ensemble constitué de la masse mobile et des deux électrodes mobiles reliées électriquement au troisième pôle au travers des ressorts de suspension comme le montre la Figure 4.10. Il existe dès lors plusieurs manières d'exciter ce résonateur suivant que l'on veuille un fonctionnement en moteur linéaire, en filtre ou en oscillateur [19] [20].

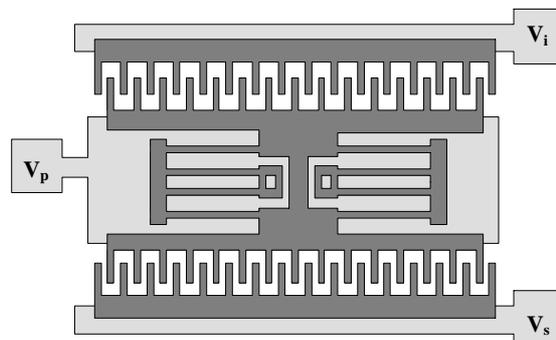


Figure 4.10. Connexions électriques d'un micro-résonateur en peigne.

Au niveau technologique, il existe plusieurs implémentations possibles de ce composant.

La Figure 4.11 montre des images MEB de micro-résonateurs fabriqués avec la technologie de micro-usinage en surface MUMPS. La disposition la plus courante est à déplacement longitudinal mais il existe également des micro-résonateurs à déplacement rotatif [20].

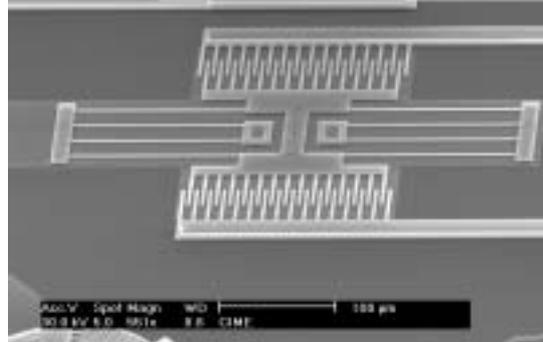


Figure 4.11. Image MEB de micro-résonateurs en peigne, à déplacement longitudinal.

- **Formulation mathématique**

L'électrode fixe et l'électrode mobile, interdigitées, forment une capacité variable C . Soumise à une différence de potentiel V l'expression de l'énergie potentielle emmagasinée dans cette capacité est :

$$(eq.16) \quad E = \frac{1}{2} CV^2$$

La force électrostatique dérive de l'énergie potentielle, on a alors la relation suivante :

$$(eq.17) \quad \vec{F} = -grad(E)$$

Selon l'axe x cette équation devient :

$$(eq.18) \quad F_x = -\frac{\partial(\frac{1}{2} CV^2)}{\partial x} = -\frac{1}{2} V^2 \frac{\partial C}{\partial x}$$

En ne prenant en compte que les capacités latérales (surfaces grisées sur la Figure 4.9) et en négligeant la capacité du bout on peut écrire :

$$(eq.19) \quad C = 2\epsilon_0 \frac{e(\delta + x)}{g}$$

avec : e l'épaisseur, g l'entrefer et δ la longueur de recouvrement.

En combinant les équations (eq.18) et (eq.19) on obtient l'expression de la force électrostatique selon l'axe x :

$$(eq.20) \quad F_x = -\frac{e\epsilon_0 V^2}{g}$$

On remarque alors que la force électrostatique ne dépend pas de la position relative du doigt δ mais est directement proportionnelle au carré de la différence de potentiel appliquée aux électrodes. C'est le principal intérêt de cette forme de capacité.

Pour exciter en vibration la masse mobile, cette dernière doit être polarisée par un potentiel fixe V_p de forte valeur (typiquement de 50 à 100 V) de manière à ce que la force électrostatique soit assez conséquente pour faire bouger la masse. En imposant une tension

AC V_i à l'électrode fixe, il est possible d'exciter la masse mobile sur sa fréquence de résonance. La tension V s'écrit alors $V=V_p-V_i$

Lorsque les deux électrodes fixes sont portées à des potentiels DC équivalents les forces électrostatiques créées par la forte polarisation de la masse mobile s'annulent. L'expression de la force électrostatique d'excitation sur la masse mobile créée par l'électrode d'excitation peut alors se ramener à :

$$(eq.21) \quad F_x = -\frac{e\epsilon_0(V_p - V_i)^2}{g} \approx 2\frac{e\epsilon_0}{g}V_p.V_i$$

On remarque que la force électrostatique appliquée à la masse mobile est linéairement dépendante de la tension V_i

La structure mobile subit également l'attraction électrostatique suivant les axes y et z , de par la construction la résultante des forces en y est nulle mais la structure mobile est en position instable, cependant la compliance mécanique de la structure évite les déplacements selon l'axe y . Selon l'axe z , la force électrostatique tend à lever la structure, c'est ce que l'on appelle l'effet de lévitation.

Une fois excitée, la structure mobile va se comporter comme un oscillateur harmonique que l'on peut formuler mathématiquement par l'équation suivante :

$$(eq.22) \quad M_{eq} \frac{d^2x}{dt^2} + \frac{\sqrt{M_{eq}k}}{Q} \frac{dx}{dt} + kx = F$$

Avec :

- k la constante de raideur des ressorts définie par :

$$(eq.23) \quad k = 2Ee\left(\frac{W}{L}\right)^3$$

W et L étant respectivement la largeur et la longueur des poutres de torsion et E le module d'Young du matériau (indiqués sur la Figure 4.9).

- M_{eq} est la masse sismique équivalente. Etant donné que les ressorts travaillent en déformation et qu'ils contribuent à la masse en mouvement une expression approchée de la masse équivalente est :

$$(eq.24) \quad M_{eq} = M_p + \frac{1}{4}M_t + \frac{12}{35}M_b$$

Q le facteur de qualité à la résonance qui est lié au frottement fluide dans l'air. Une expression de ce facteur dérivant des modèles de frottement fluide de Couette et Stokes concernant le frottement de deux surfaces planes en regard est :

$$(eq.25) \quad \frac{1}{Q} = \frac{\mu}{\sqrt{kM_{eq}}} \left(\frac{\alpha A_q}{e\left(1 + \frac{e}{\delta}\right)} + \frac{A_c}{e} \right) + \frac{1}{Q_n}$$

avec : μ la viscosité de l'air, A_q et A_c sont les surfaces de recouvrement horizontales et verticales.

La pulsation propre de ce système du deuxième ordre est alors définie par :

$$(eq.26) \quad \omega_0 = \sqrt{\frac{k}{M_{eq}}}$$

Au niveau de l'électrode de sortie, le mouvement de l'électrode mobile et sa polarisation imposent un courant dans la capacité :

$$(eq.27) \quad I = \frac{dQ}{dt} = \frac{d}{dt}(CV)$$

Où V est la tension à l'électrode de sortie et vaut $V=V_p-V_s$ avec $V_p \gg V_s$. On peut écrire :

$$(eq.28) \quad I = V_p \frac{2e\epsilon_0}{g} \frac{dx}{dt} - C_0 \frac{d}{dt} V_s$$

C_0 étant la composante fixe de la capacité de l'électrode et V_s la tension de l'électrode fixe. Le courant de sortie est alors constitué d'une composante provenant du mouvement de la masse mobile et d'une composante fixe provenant de la capacité fixe entre l'électrode de sortie et la masse mobile.

• **Analogie Electrique Mécanique**

Le développement de circuits équivalents repose sur l'analogie de la description mathématique des phénomènes électriques et mécaniques (incluant l'acoustique).

Cette analogie résulte de la similitude des équations différentielles régissant les circuits électriques ou les montages mécaniques [18] [19] [20].

Par exemple l'équation de la dynamique en mécanique stipule que : $F = m\gamma = m\dot{u} = m\ddot{x}$ alors qu'en électricité la différence de potentiel aux bornes d'une bobine de self- induction L s'écrit $v = L\dot{i} = L\ddot{q}$. Dans cette approche F joue le rôle de v, tandis que x est l'analogie mécanique de la charge q. Pour les autres grandeurs mises en jeu les équivalences sont résumées dans le tableau 4.4 ci-dessous. Cette analogie est appelée analogie directe.

<i>Grandeurs mécaniques</i>	<i>Grandeurs électriques</i>
Force F	Tension v
Vitesse $u = \dot{x}(t)$	Courant $i = \dot{q}(t)$
Déplacement x	Charge q
Quantité de mouvement p	Flux λ
Masse m	Inductance L
Souplesse 1/k	Capacité C
Amortissement visqueux	Résistance R

Tableau 4.4. Analogie directe électromécanique.

Cette analogie implique que la topologie des circuits électriques et mécaniques est différente. Par exemple un circuit "série" en mécanique aura pour équivalent un circuit "parallèle" électrique.

Dans la suite, nous utiliserons essentiellement cette analogie directe. Il faut noter par ailleurs qu'il existe une autre analogie possible, appelée analogie inverse qui relie F à i, v à u, m à C, etc.

4.4 Simulation globale de microsystemes

Les *Systems-on-Chip* (SoC) et les *Systems-on-Package* (SoP) deviennent de plus en plus des systèmes hétérogènes. Des parties numériques, des parties analogiques, les dispositifs de communication et les composants de microsystemes sont intégrés sur la même puce. Afin de valider des systèmes contenant une telle variété de composants, la simulation globale multi-domaine de différents aspects devient particulièrement importante.

Un environnement de simulation mixte permet la validation des systèmes complexes multi-domaine contenant sur la même puce : les parties numériques (Processeur, RAM, ROM), les parties analogiques (convertisseurs, filtres), les dispositifs de communication (antennes RF) et les composants de microsystemes (des capteurs et des actionneurs) [13] [14].

L'utilisation des langages de haut niveau d'abstraction tel que C et VHDL est essentielle pour la conception et la validation du logiciel et du matériel numérique. En outre, les langages comportementaux tels que HDL-A, Verilog-A et MATLAB permettent de modéliser les sous-systèmes analogiques et les dispositifs de microsystemes [11] [12]. L'ensemble complet de modèles pour le logiciel, les composants électroniques et les dispositifs microsysteme peut alors être analysé dans un environnement de co-simulation. De plus, un tel environnement permet la simulation de défauts des dispositifs de microsystemes et l'évaluation thermique du système matériel/logiciel.

Le Figure 4.12 montre le flot dans un environnement de co-simulation. Dans cet environnement l'approche multi-langage / multi-simulateur est adoptée pour la description à haut niveau de conception. Cette approche reflète l'hétérogénéité des systèmes sur la puce et ainsi la diversité des outils de CAO utilisés dans la conception d'un tel système.

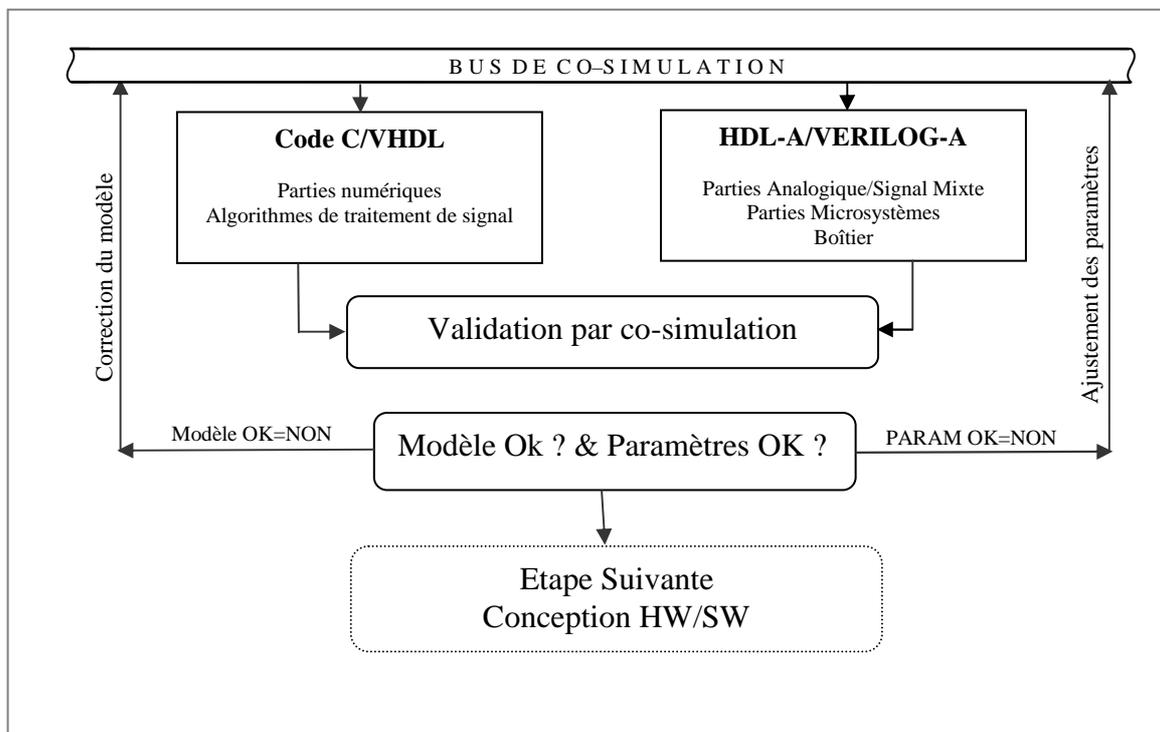


Figure 4.12. Environnement de co-simulation.

Un tel flot de conception permet la validation d'un SoC hétérogène au plus haut niveau de conception ce qui représente des réductions considérables au niveau de temps de conception [26]. De plus, cette approche de simulation doit pouvoir produire des résultats avec exactitude si les modèles sont correctement choisis. Enfin cette méthodologie permet l'exploration d'une architecture pour prendre des décisions sur une conception spécifique [7] [8].

Pour illustrer la méthodologie de co-simulation microsystemes, nous avons choisi comme exemple le multiplexeur de fibres optiques à micro-miroirs. C'est un système hétérogène aux divers composants qui se compose de trois parties essentielles [1] [5] :

- La partie numérique responsable des commandes : il calcule des ordres électroniques (tension) qui commandent les micro-miroirs mécaniques composant la partie optique.
- Une partie analogique représentée par un convertisseur électromécanique qui transforme la tension de sortie de la partie commande en ordres mécaniques, en termes de distance, pour chacun des miroirs.
- Une partie optique qui se compose d'une matrice de miroirs 4x4, 4 sources de lumière (G1, G4), 4 détecteurs de poutre (D1, D4) et 8 objectifs (L1-L8). Les objectifs L1 et L4 sont utilisés pour aligner les faisceaux venant des fibres d'entrée tandis que les objectifs L4 et L8 sont utilisés pour focaliser des signaux dans les fibres de sortie.

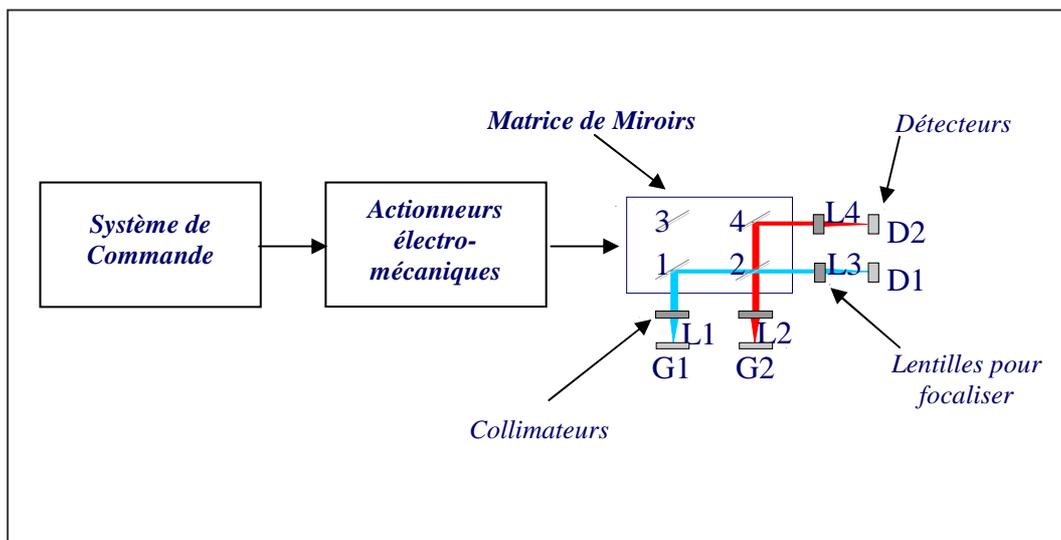


Figure 4.13. Multiplexeur optique à micro-miroirs.

Dans la Figure 4.13, l'opération de commutation est réalisée par le mouvement mécanique des miroirs orientant la circulation de données des entrées (G1 à G4) aux sorties désirées (D1 à D4). Le couplage optique entre les fibres d'entrée et les fibres de sortie peut être mesuré et utilisé comme entrée au système de commande. Le système comporte une unité centrale de traitement qui contient l'algorithme de commande; des convertisseurs analogique/numérique et numérique/analogique; circuits de traitement de signal; parties électromécaniques et circuits optiques [10] [14]. Ceci est illustré sur la Figure 7.8

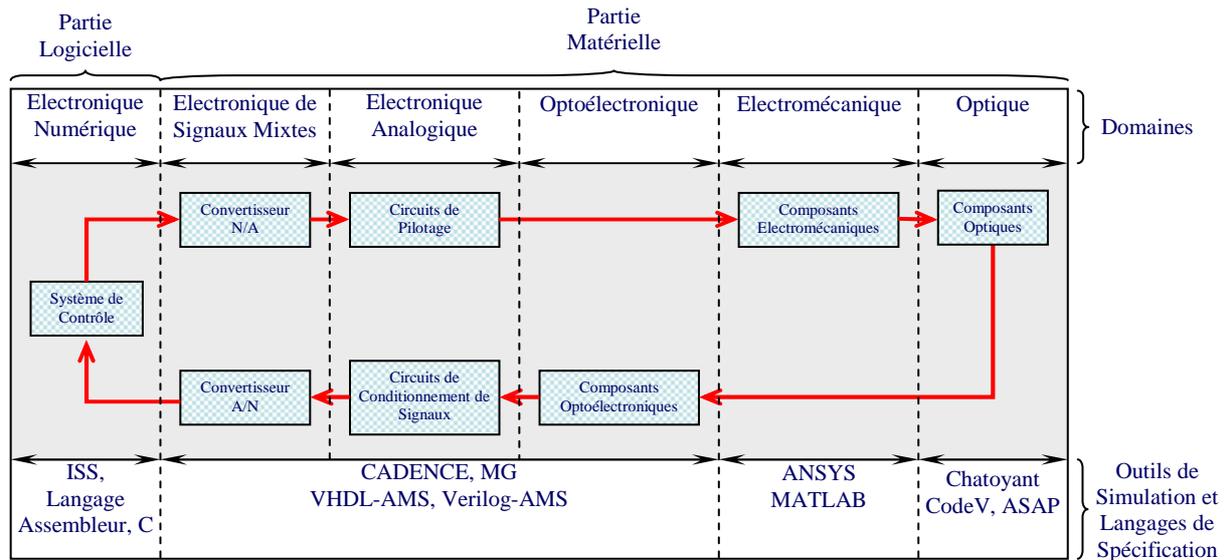


Figure 4.14. Architecture d'un multiplexeur à micro-miroir

Une simulation globale d'un tel système implique une description multi-langage (SystemC/MatLab/C++) [4] et un environnement multi-domaine de co-simulation (numérique/analogue/mécanique/optique). Le système de commande est implémenté en SystemC et la partie électromécanique dans Matlab. Pour ce qui concerne les circuits optiques, miroirs, objectif, générateurs de faisceau et détecteurs), des modèles de C++ des bibliothèques de Chatoyant [10] [15] [16] ont été employés.

Les spécifications du système sont présentées sur la Figure 4.14. Pour la simplicité de l'explication, seulement deux miroirs de la matrice des micro-miroirs sont montrés dans la figure. Les modules dans le système communiquent par des canaux de transmission qui encapsulent des protocoles simples de *Handshaking*. Chaque module de miroir reçoit les commandes de commutation du convertisseur électromécanique avec un protocole de transmission *FIFO*. Pour spécifier l'interface entre les modules optiques et ceux de canaux de transmission, les modules ont des ports internes et externes. Comme représenté sur la Figure 4.15, les ports internes sont des ports FIFO et les ports externes reliés aux canaux de transmission sont des ports simples de Handshaking.

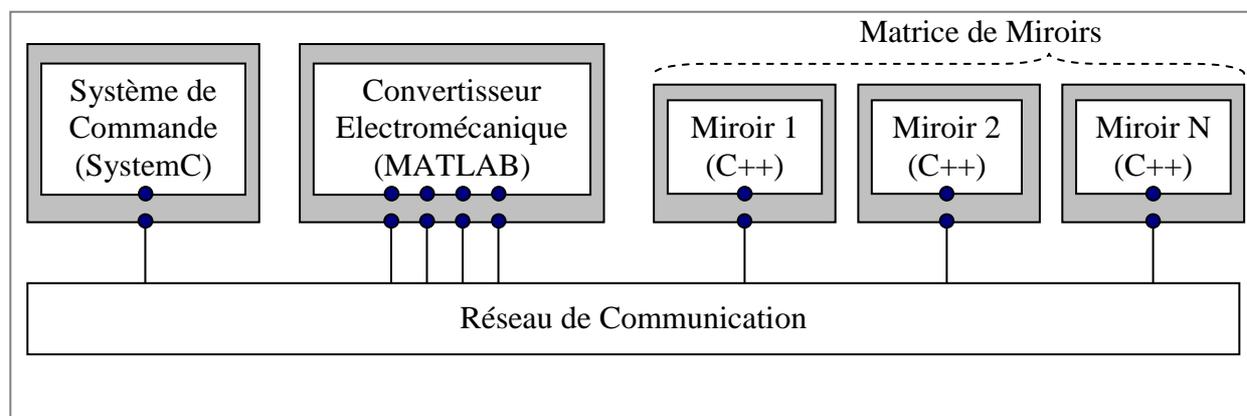


Figure 4.15. Architecture d'un multiplexeur à micro-miroir.

La méthodologie présentée a été employée pour la génération du modèle du système, ensuite la co-simulation globale de ce système a été exécutée, en utilisant une rangée de miroirs 2x2

qui ont des entrées de deux générateurs, des sorties de deux faisceaux, et deux détecteurs de lumières. Les résultats de co-simulation sont illustrés dans la Figure 4.16 :

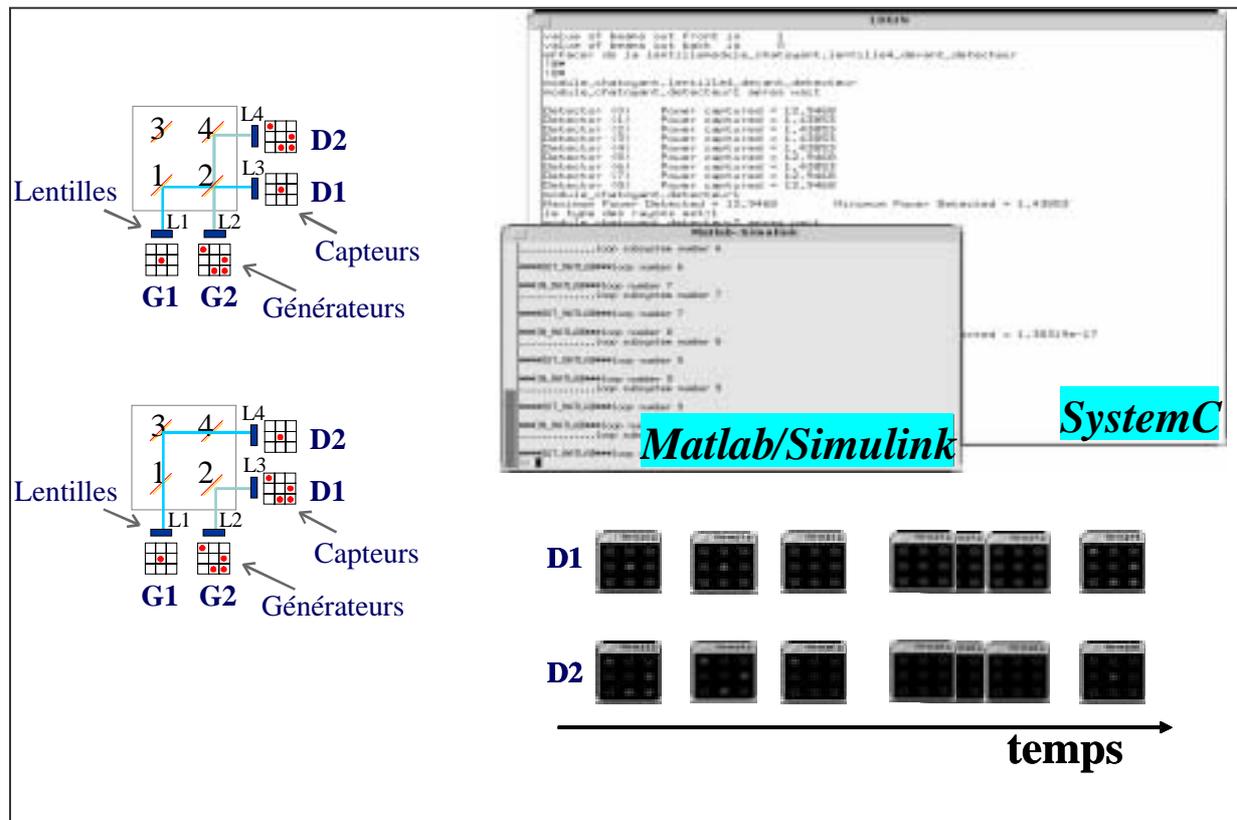


Figure 4.16. Co-Simulation de commutation

La co-simulation du commutateur optique a été effectuée sur une station SUN Ultra Sparc 1. Le temps de simulation était environ 30 secondes. La Figure 4.16 montre l'évolution des signaux reçus par les photodétecteurs au fur et à mesure que la co-simulation progresse. Initialement, le miroir 1 réfléchit le faisceau provenant de G1 vers D1 alors que le miroir 4 réfléchit le faisceau provenant de G2 vers D2. Pendant la simulation, les miroirs changent graduellement leurs positions en conformité avec les ordres envoyés par le système de commande [10].

4.5 Conclusion

Dans ce chapitre, nous avons montré les différentes méthodologies de la modélisation des microsystèmes en adoptant deux niveaux d'abstraction pour aborder ce problème :

- Niveau composant : nous avons montré l'importance de l'utilisation des techniques de description par les blocs comportementaux (*behavioral*) et des éléments de circuit localisé. Nous avons ensuite utilisé ces techniques pour modéliser et simuler un composant microsystème (le convertisseur électrothermique).
- Niveau système : nous avons montré la nécessité de simuler un ensemble de composants très hétérogènes autour d'un microsystème, incorporant dans la même puce du matériel et du logiciel. Nous avons défini une méthodologie de co-simulation de microsystèmes. Cette méthodologie profite directement de l'expérience des

différents groupes de travail dans l'utilisation des langages et des simulateurs spécialisés dans chaque domaine. Nous avons validé cette méthodologie sur la simulation globale d'un système de commutateur optique.

Nous pensons que l'environnement de la synthèse et d'optimisation (développé à travers le chapitre 6) complète l'écart entre les deux niveaux (composant et système).

Références du chapitre 4

- [1] C.A. Brebbia, J.C.F. Telles, L.C. Wrobel, "Boundary Element Techniques", Springer-Verlag, 1984.
- [2] M. Gyimesi, and D. Ostergaard, "Electro-Mechanical Transducer for MEMS in ANSYS", in *Proceeding 1999, International Conference on Modeling and Simulation of Microsystems*, 19-21 April 1999.
- [3] M.J. Fagan, "Finite Element Analysis, Theory and Practice", Longman Scientific and Technical, 1992.
- [4] D. Ostergaard, and M. Gyimesi, "Finite element based reduced order modeling of micro electro mechanical systems (MEMS)", In *Modeling and simulation of MEMS*, 2000.
- [5] B. Romanowicz, M. Laudon, P. Renaud, H. P. Amman, A. Boegli, V. Moser, and F. Pellandini, "Modeling and simulation of electromechanical transducers in microsystems using an analog description language", in *Proceedings. European Design and test Conference. ED&TC*, pp. 119-123, Paris, France, March 1997.
- [6] H. H. Woodson and J. R. Melcher, *Electromechanical dynamics*, vol. I, II and III, Wiley, New York, NY, USA, 1968.
- [7] F. Balarin, *Hardware-Software Co-design for Embedded Systems*, Boston: Kluwer Academic Publishers, 1997.
- [8] B. Courtois, J. M. Karam, S. Mir, M. Lubaszewski, V. Székely, M. Rencz, G. Kelly, J. Alderman, A. Morrissey, K. Hofmann, M. Glesner, "CAD, CAT and MPW for MEMS", *Workshop on Synthesis and System Integration of Mixed Technologies, SASIMI'98*, pp. 207-219, Sendai, Japan, October 1998.
- [9] K. Hinnes, G. Boriello, "Dynamic communication models in embedded systems co-simulation", *Design Automation Conference, DAC'97*, pp. 395-400, June 1997.
- [10] L. Kriaa, W. Youssef, G. Nicolescu, S. Martinez, S. Levitan, J. Martinez, T. Kurzweg, A. A. Jerraya, B. Courtois. "SystemC-based cosimulation for global validation of MOEMS", in *Design, Test, Integration and Packaging of MEMS/MOEMS, DTIP'02*, B. Courtois, editor, Cannes, May 6-8, 2002.
- [11] P. L. Marrec, "Cosimulation multiniveaux dans un flot de conception multilangage", Thèse de Doctorat, Ecole Doctorale EEATS, Institut National Polytechnique de Grenoble, juin 2000.
- [12] G. Nicolescu, S. Martinez, L. Kriaa, W. Youssef, S. Yoo, B. Charlot, A. Jerraya, "Application of multi-domain and multi-language cosimulation to an optical MEM switch design", *Asia South Pacific Design Automation Conference, ASPDAC'02*, Bangalore, India, January 7-11, 2002.

- [13] S. D. Senturia, "CAD challenges for microsensors, microactuators, and microsystems", *Proceedings of the IEEE*, vol. 86, no. 8, pp. 1611-1626, 1998.
- [14] Z. Juneidi, K. Torki, G. Nicolescu, S. Martinez, B. Courtois, A. A. Jerraya, "Global modeling and simulation of systems on chip embedding MEMS devices", *ASICON 2001*, shanghai, China, 2001.
- [15] T. P. Kurzweg, S. P. Levitan, P. J. Marchand, J. A. Martinez, K. R. Prough, D. M. Chiarulli, "CAD for optical MEMS", *Design Automation Conference, DAC'99*, New Orleans, USA, June 20-25, pp. 879-884, 1999.
- [16] S. P. Levitan, T. P. Kurzweg, P. J. Marchand, M. A. Rempel, D. M. Chiarulli, J. A. Martinez, J. M. Bridgen, C. Fan, F. B. McCormick, "Chatoyant: a computer-aided-design tool for free-space optoelectronic systems", *Applied Optics*, vol. 37, no. 26, pp. 6078-6092, 1998.
- [17] Z. Juneidi, K. Torki, B. Charlot, B. Courtois, "MEMS Synthesis and Optimization", *Design, Test, Integration and packaging of MEMS/MOEMS*, Cannes, France, April 2001.
- [18] W.C. Tang, T.-C.H. Nguyen, M.W. Judy and R. T. Howe, "Electrostatic-comb Drive of Lateral Polysilicon Resonators", *Transducers '89*, Vol. 2, pp. 328-331, June 1990.
- [19] T. Hirano, T. Furuhashi, K.T. Gabriel and H. Fijita, "Design, Fabrication and Operation of Submicron Gap Comb-Drive Microactuator", *J. of MEMS*, Vol. 1, No. 1, pp. 52-9, March 1992.
- [20] W.C. Tang, M.G. Lim and R.T. Howe, "Electrostatically Balanced Comb Drive for Controlled Levitation", *Technical Digest IEEE Solid-State Sensor and Actuator Workshop*, pp. 23-27, June 1990.

5

Dessin du masque et règles de dessin pour microsystemes

5.1 Introduction

Dans une approche ascendante de la conception de microsystemes, l'étape de dessin de masque est la première étape. Il s'agit en fait de rassembler des différents motifs d'une manière créative pour réaliser le composant microsysteme désiré. Or, pour assurer le bon fonctionnement de celui-ci, un certain nombre de règles technologiques concernant les dimensions et les espacements de ces motifs doivent être respectées. Le concepteur doit de plus s'assurer que les interconnexions entre les différents types de composants ne dégradent pas trop les performances du système. Enfin, pour des raisons de coût, il doit aussi faire en sorte que le système réalisé soit le plus compact possible de façon à économiser au maximum la quantité de silicium requise. Une étape de vérification après coup reste malgré tout nécessaire pour s'assurer que toutes les règles de dessin ont bien été respectées et que le redimensionnement du système n'aura pas d'influence critique sur le fonctionnement. Etant issus de la famille des outils de CAO pour la microélectronique, les outils de CAO pour microsystemes ont hérité des mêmes handicaps que leurs prédécesseurs.

5.2 Les Formes Non-Manhattan

La plupart des outils CAO pour microélectronique (et par conséquence pour microsystemes) fonctionnent avec les polygones rectangulaires [9] [10] [11] (dites les formes Manhattan, en anglais les *Manhattan shapes*). Ce sont des polygones avec des angles multiples de 90 degrés, voir la Figure 5.1 :

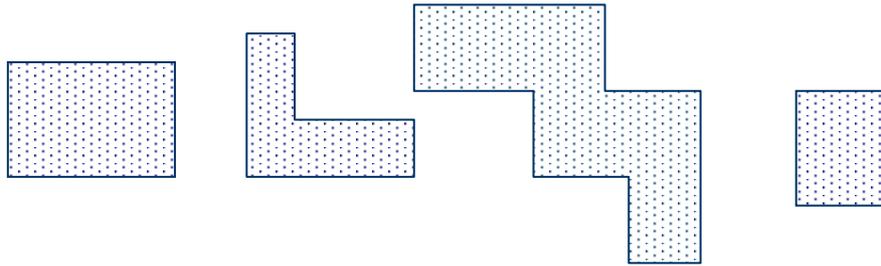
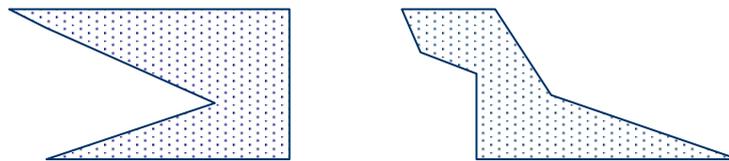
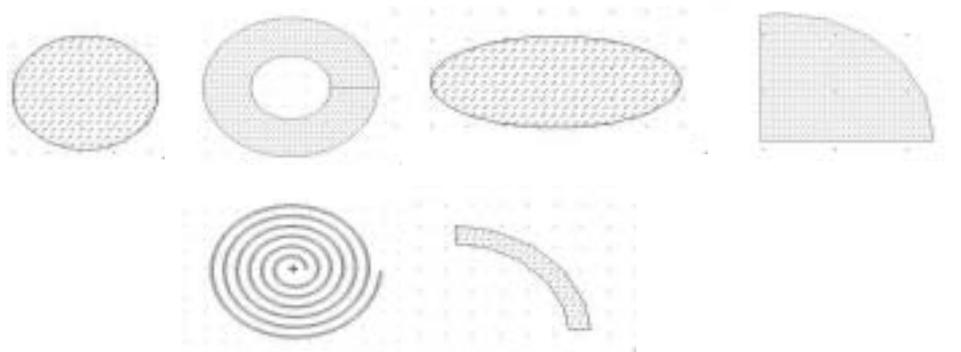


Figure 5.1. *Les formes Manhattan*

En microélectronique, les polygones avec des angles (0, 90, 180) sont suffisants pour concevoir n'importe quel dispositif. Ce n'est pas le cas pour les microsystemes, où le besoin des formes Non-Manhattan est nécessaire la plupart du temps. Les formes Non-Manhattan sont soit des polygones avec des angles libres, soit des formes coniques. La Figure 5.2 (a) et (b) montre différents types de forme Non-Manhattan.



(a) *Les polygones Non-Manhattan*



(b) *Les Formes Coniques*

Figure 5.2. *Les formes Non-Manhattan.*

Dans la pratique, il n'existe pas d'outil CAO capable d'élaborer des formes Non-Manhattan [10] [11]. En utilisant des procédures d'échantillonnage, les éditeurs de dessin de masques transforment les formes Non-Manhattan en polygones composés d'un grand nombre de sommets et de segments.

La Figure 5.3 montre le résultat d'une transformation par approximation d'une spirale par des polygones.

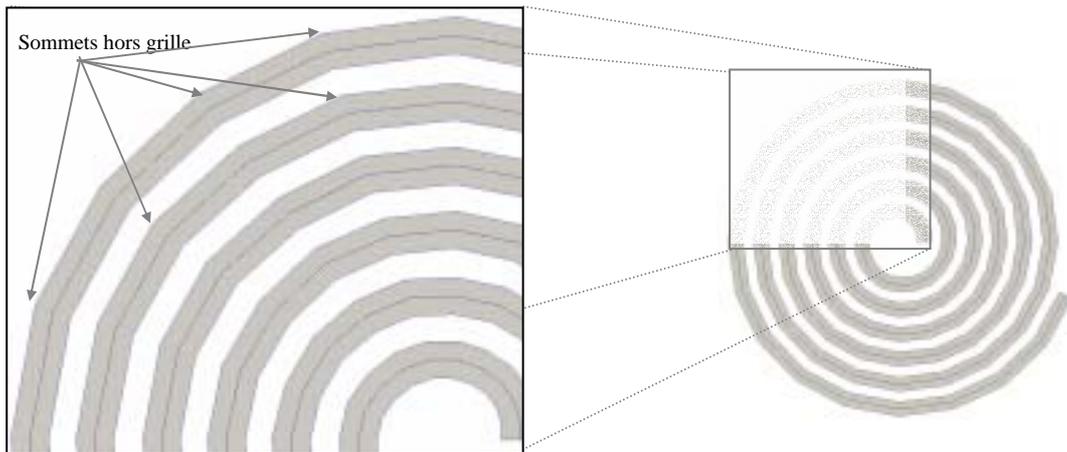


Figure 5.3. Echantillonnage d'une spirale en polygones.

Les résultats directs d'un tel échantillonnage sont d'avoir des sommets hors grille, ou d'avoir un échantillonnage non-uniforme de polygones (suite à un *snap to grid*). La Figure 5.4 (a) montre un échantillonnage hors grille et la Figure 5.4 (b) montre un échantillonnage non-uniforme.

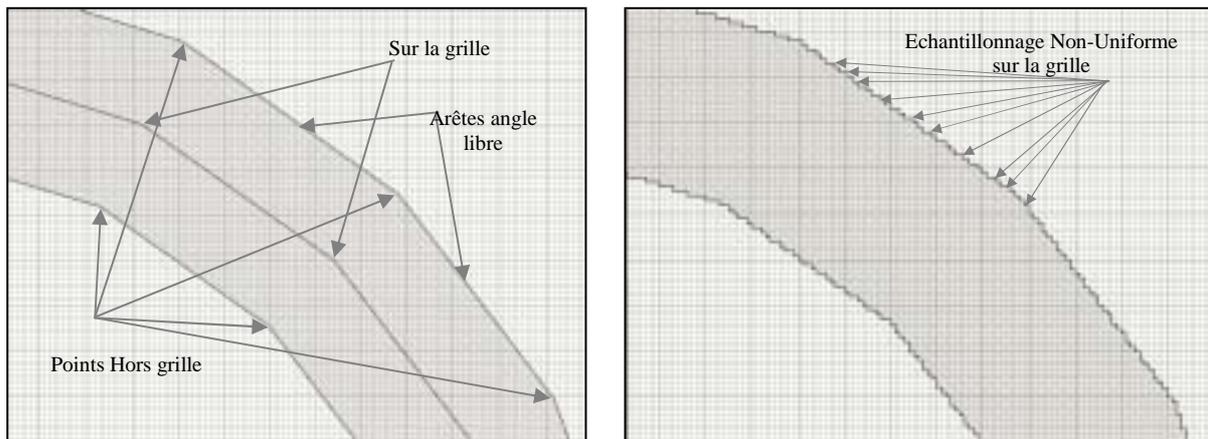


Figure 5.4. (a) Echantillonnage hors grille, (b) Echantillonnage non-uniforme.

5.3 La discrétisation des formes Non-Manhattan

La question que nous nous posons ici est de savoir comment doit s'effectuer le passage de l'espace analogique (formes continues non-échantillonnées) à l'espace discret (forme non continues échantillonnées et *snaped to grid*) de manière que la discrétisation d'une courbe continue soit approchée par une courbe discrète, où la discrétisation consiste à représenter par des points de grille une entité analogique.

Cette question soulève un problème à double sens :

- Etant donné une entité continue (forme non échantillonnée), comment construire l'entité discrète correspondante et quelles sont les propriétés de cette entité discrète par rapport à celles de l'entité initiale ?
- Etant donné une entité discrète, quelle est (ou quelles sont) la (ou les) entité(s) continue(s) dont elle peut provenir ?

5.3.1 Fondements mathématiques

Dans un premier temps, nous apportons des réponses à ces deux questions dans le cas de la discrétisation d'une courbe (forme conique) § 5.3.1.1. Nous nous intéresserons ensuite plus particulièrement aux segments de droite (polygone d'angle libre) § 5.3.1.2.

5.3.1.1 Discrétisation d'une courbe

Du point de vue analogique, les arcs et courbes sont des ensembles de mesures nulles, qui séparent totalement l'espace en deux régions distinctes. Aussi il s'agit de trouver une représentation discrète de ces entités analogique qui vérifient les mêmes propriétés.

Différentes approches du problème de la discrétisation d'une courbe sont possibles :

I. Discrétisation par pavés semi-ouvert

Un point de la grille est retenu si l'intersection entre le semi-ouvert qui lui est associé et la courbe à discrétiser est non vide [1], voir la Figure 5.5 :

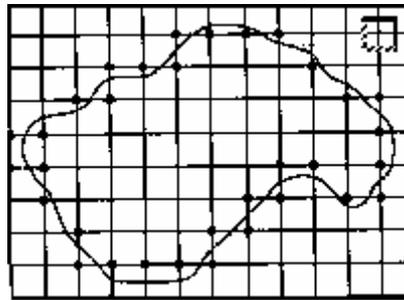


Figure 5.5. Discrétisation par pavés semi-ouverts.

II. Discrétisation par intersection avec la grille

La nature même d'un arc ou d'une courbe permet de développer un procédé fondé sur les intersections entre la courbe et les arrêtes de la grille [2] [3].

Pour tout point de la courbe de coordonnées (x,y) tel que x ou y est entier, on procède de la manière suivante :

Si $x=i$ entier, alors j est défini par $j-1/2 < y \leq j+1/2$

Si $y=j$ entier, alors i est défini par $i-1/2 < x \leq i+1/2$

Autrement dit pour x ou y entier, donc en tout point où la courbe coupe les arrêtes de la grille, le point (x,y) est représenté par le point de grille le plus proche, (voir la Figure 5.6).

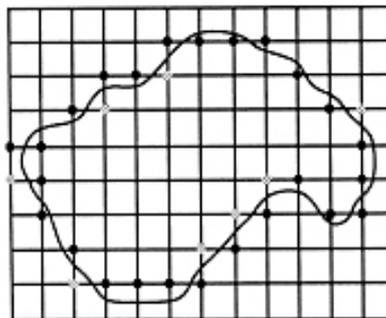


Figure 5.6. Discrétisation par intersection avec la grille, montrant les points supprimés par rapport au résultat de la Figure 5.2.

III. Discrétisation par intersection avec la grille d'un même côté de la courbe

On peut reprocher aux procédés précédents de ne pas garantir que l'ensemble discret associé ne se situe pas à l'intérieur de la courbe. On peut facilement remédier à cela en orientant la courbe continue [1] [2] [3] [4].

Soit une courbe fermée simple C de \mathfrak{R} orientée dans le sens trigonométrique, la représentation discrète C est définie par :

A tout point R de C dont une coordonnée est entière (donc point d'intersection avec la grille) on associe la représentation discrète P à gauche de la courbe orientée C sur la même ligne ou colonne de la grille et tel que la distance est maximum $(P,R) < 1$.

Cette méthode revient à se déplacer sur la courbe dans le sens trigonométrique à chaque intersection avec le maillage, à affecter à C le premier point discret à gauche sur la grille (donc par déplacement horizontal ou vertical), voir la Figure 5.7.

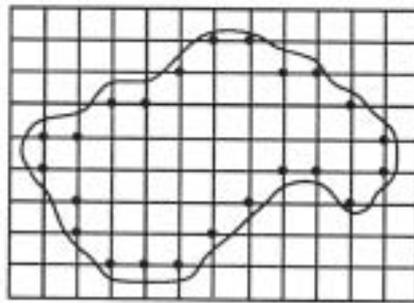


Figure 5.7. Discrétisation par intersection avec la grille d'un même côté de la courbe.

L'intérêt de cette méthode réside dans le fait que tous les points de la représentation discrète sont à l'intérieur de la courbe fermée.

5.3.1.2 Discrétisation d'un segment de droite

Pour caractériser le segment de droite discrète, la meilleure approche consiste à étudier le procédé qui permet de l'obtenir à partir d'un segment.

I. Plus proche voisin discret

Avec la méthode de discrétisation par pavé semi-ouvert associé à tout point analogique, on obtient un segment discret représenté par la Figure 5.8 [5] [6] :

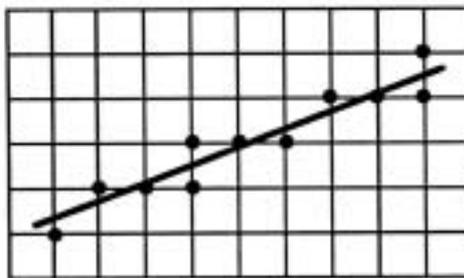


Figure 5.8. Discrétisation d'un segment par le plus proche voisin discret.

II. Plus proche voisin discret de même côté

Dans le cas du segment de droite, on peut utiliser la même remarque que pour la courbe, à savoir qu'on ne cherche pas à discrétiser le segment en tant que tel, mais à discrétiser la frontière rectiligne d'une forme (qui est ici un demi-plan) fermée. Aussi, un procédé

consiste à retenir les points discrets situés d'un même côté du segment analogique, chaque fois que celui-ci coupe une arête de grille [5] [8], voir la Figure 5.9.

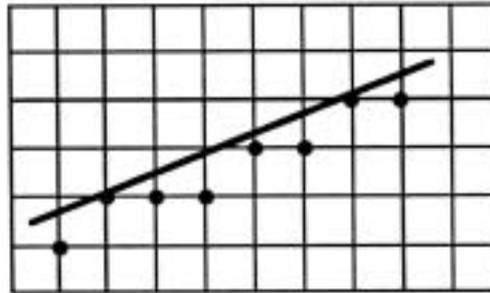


Figure 5.9. *Discrétisation d'un segment par le plus proche voisin discret de même côté.*

III. Plus proche voisin selon une direction de grille

Une autre définition également utilisée, permettant d'aboutir à un segment discret et qui est en plus positionné d'une manière équitable par rapport au segment analogique, combine les deux procédés précédents [5] [6], voir la Figure 5.10.

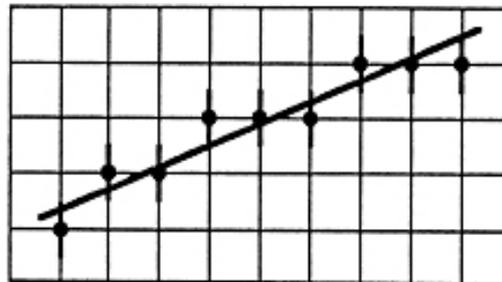


Figure 5.10. *Discrétisation d'un segment par le plus proche voisin selon la direction de la grille.*

Nous supposons le segment analogique de pente inférieure à 1.

A chaque point P de la grille est associé le segment vertical semi-ouvert en bas centré en P et de longueur 1.

Tout point du segment analogique qui appartient à un tel segment de centre P définit le point P sur le segment discret, les autres points du segment analogique ne sont pas pris en compte.

Selon ce procédé, chaque fois que le segment analogique supposé de pente < 1 coupe un trait vertical de la grille, on définit le point le plus proche de la grille sur ce trait comme faisant partie du segment discret.

Avec un tel procédé on obtient ce que l'on appelle une caractérisation syntaxique des segments de droite discrète. Cette caractérisation est fournie par les éléments suivants [7] :

- Un segment de droite discret est formé de sections. Une section étant caractérisée par une suite de taille maximale de directions identiques qui définissent la direction de la section (huit directions sont possibles).
- Les sections formant un segment de droite discret ont au plus deux directions distinctes qui diffèrent d'une unité (d'un angle de 45°).
- Pour l'une de ces directions, les sections sont toutes de longueur 1.
- Pour l'autre direction, les sections (exceptées la première et la dernière) ne peuvent avoir pour longueur que deux valeurs distinctes qui ne diffèrent que d'une unité

(soit n et $n+1$). De plus l'une de ces valeurs n'apparaît que de manière isolée, voir la Figure 5.11.

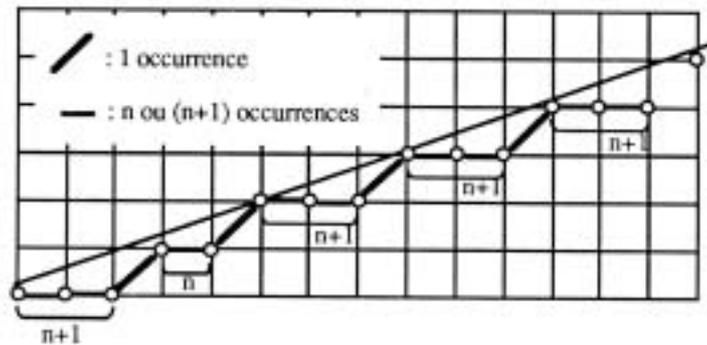


Figure 5.11. Segment de droite discrète et caractérisation syntaxique.

5.3.1.3 Discrétisation des formes coniques

Le problème se pose de la manière suivante : on a une forme conique générée par la fonction $f(x) = y$ sur \mathcal{X} , on veut trouver la représentation discrète (échantillonnage uniforme sur les points de grille) de cette forme.

Pour ramener le problème à son cas initial on suppose que la forme est centrée comme dans la Figure 5.12 :

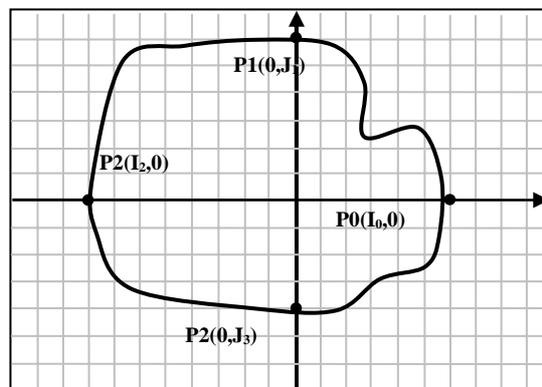


Figure 5.12. Discrétisation de forme conique : cas initial.

On veut appliquer la méthode exposée dans le paragraphe précédent, pour cela on divise la forme en quatre courbes qui sont réparties sur les quatre quarts de l'espace des coordonnées, ce qui nous permet d'orienter les quatre courbes dans le sens trigonométrique comme le suivant.

- 1- courbe 1 dans le premier quart orienté de $P0 \Rightarrow P1$.
- 2- courbe 2 dans le deuxième quart orienté de $P1 \Rightarrow P2$.
- 3- courbe 3 dans le troisième quart orienté de $P2 \Rightarrow P3$.
- 4- courbe 4 dans le quatrième quart orienté de $P3 \Rightarrow P4$.

Cette procédure est décrite comme par l'algorithme

```

Q1 :Pour j= 0 à j= J1
    Début
        Coordonnée x de Pi,j est i tel que  $i \leq f^{-1}(j) < i+1$ 
        Coordonnée y de Pi,j est j
    Fin
    Pour i=I0 à i= 0
        Début
            Coordonnée x de Pi,j est i
            Coordonnée y de Pi,j est j tel que  $j \leq f(i) < j+1$ 
        Fin
Q2 :Pour i= 0 à i= I2
    Début
        Coordonnée x de Pi,j est i
        Coordonnée y de Pi,j est j tel que  $j \leq f(i) < j+1$ 
    Fin
    Pour j=J1 à j= 0
        Début
            Coordonnée x de Pi,j est i tel que  $i+1 \leq f^{-1}(j) < i$ 
            Coordonnée y de Pi,j est j
        Fin
Q3 :Pour i= I2 à i=0
    Début
        Coordonnée x de Pi,j est i
        Coordonnée y de Pi,j est j tel que  $j+1 \leq f(i) < j$ 
    Fin
    Pour j=0 à j= J3
        Début
            Coordonnée x de Pi,j est i tel que  $i+1 \leq f^{-1}(j) < i$ 
            Coordonnée y de Pi,j est j
        Fin
Q4 : Pour i=0 à i= I0
    Début
        Coordonnée x de Pi,j est i
        Coordonnée y de Pi,j est j tel que  $j \leq f(i) < j+1$ 
    Fin
    Pour j= J3 à j=0
        Début
            Coordonnée x de Pi,j est i tel que  $j+1 \leq f^{-1}(i) < j$ 
            Coordonnée y de Pi,j est j
        Fin
    Fin

```

Algorithme 5.1. L'algorithme de discrétisation d'une forme conique.

Cet algorithme peut être utilisé pour les générateurs de dessin de masques ainsi que pour effectuer un *snap to grid* uniforme pour les formes coniques.

5.3.1.4 Discrétisation des polygones avec l'angle libre (Non-Manhattan)

Le problème de l'échantillonnage d'un polygone Non-Manhattan peut être ramené dans le problème de discrétisation de segment droit puisque un tel polygone est composé d'une séquence de segments droits. On adopte la méthode expliquée dans le §.5.3.1.1 (III) qui est la technique « plus proche voisin selon la direction de la grille ».

Notre algorithme fonctionne selon la pente du segment droit, on peut distinguer deux cas :

- 1- La pente est inférieure à 1 et supérieure à -1 comme dans la Figure 5.13 suivante.

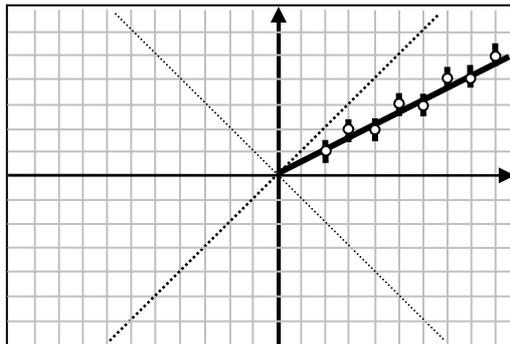


Figure 5.13. Discrétisation d'un segment dont la pente est <1 et >-1 .

a)- à chaque point P de la grille est associé le segment vertical semi-ouvert en bas centré en P et de longueur 1.

b)- tout point de segment analogue et qui appartient à un tel segment de centre P définit le point P sur le segment échantillonné.

2- La pente est supérieure à 1 ou inférieure à -1 :

Dans ce cas on procède de la même manière en associant des segments horizontaux à la place des segments verticaux semi- ouverts à gauche pour chaque point de la grille, voir la Figure 5.14 :

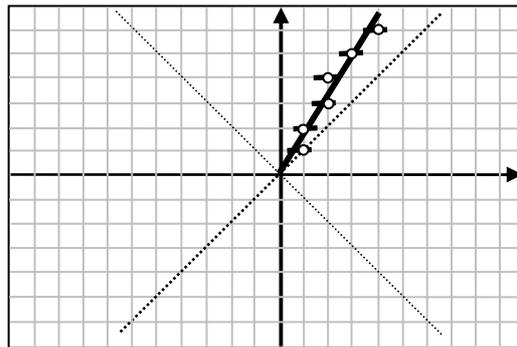


Figure 5.14. Discretisation d'un segment dont la pente est >1 ou <-1 .

L'algorithme est le suivant :

```

Si  $-1 < \text{pente} < 1$ 
Début
    Pour i entier
        Coordonnée x de  $P_{i,j} = i$ 
        Coordonnée y de  $P_{i,j} = i$  tel que  $j-1/2 < f(i) \leq j+1/2$ 
    Fin
Si pente  $> 1$  ou pente  $< -1$ 
Début
    Pour j entier
        Coordonnée x de  $P_{i,j}$  est i tel que  $i-1/2 < f^{-1}(j) \leq i+1/2$ 
        Coordonnée y de  $P_{i,j}$  est j
    Fin
    
```

Algorithme 5.2. L'algorithme de discretisation d'un segment droit.

5.4 Les règles de dessin pour microsystemes

Les travaux concernant les règles de dessin pour microsystemes, ont été effectués dans le cadre du projet européen TALENT. Pendant ce projet, nous avons été chargés de l'implémentation des modules de vérification de règles de dessin pour trois kits de conception. Ces trois kits sont SensoNor, Bosch et AMS Engineering Kits.

Comme dans le monde de la microélectronique, les procédés de fabrication de composants microsystemes sont limités par plusieurs aspects [19]. D'abord il y a un seuil de précision que ces procédés ne peuvent dépasser, ce qui reflète l'incapacité à résoudre les petites lignes et les espacements étroits. Ensuite ces procédés sont limités par l'habilité de contrôler les processus chimiques vitaux pour suspendre les structures de type poutre ou membrane. Ces limitations définissent un maximum de seuil de précision auquel on peut ajuster un design microsysteme sans que cela se passe au détriment de la fiabilité du design.

Les limitations d'un procédé technologique microsysteme sont traduites sous la forme des règles de dessin. Les règles de dessin d'un procédé microsysteme sont communiquées d'un coté au concepteur microsysteme pour lui permettre de concevoir ces composants microsystemes dans les limites de la technologie, et de l'autre coté au développeur d'outils de vérification de masque pour que l'expérience acquise par le concepteur en utilisant ces règles de dessin soit exploitée dans l'implémentation d'outils de vérification de masques plus performants [17].

Un des problèmes les plus persistants auquel le développeur des outils de vérification de masques est confronté, est de réduire le nombre d'erreurs fausses, et non vérifiées. Une fausse erreur est une erreur signalée par l'outil de vérification du masque où il n'y a pas vraiment de violations aux règles de dessin. La plupart des outils de vérification de masque souffrent de ce problème ce qui est illustré dans la Figure suivante [17] [19] :

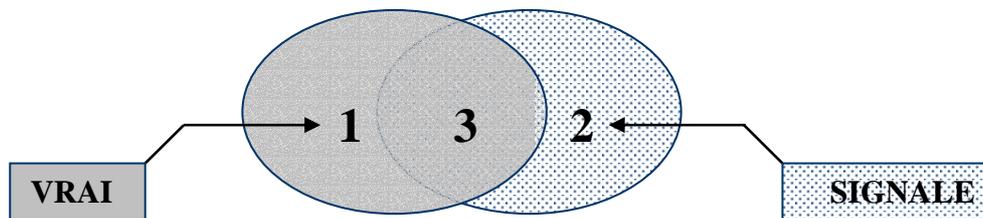


Figure 5.15. Les fausses et les vraies signalisations.

La région 3 présente les vraies violations de règles de dessin qui ont été bien signalées par l'outil de vérification. La région 1 présente les vraies violations qui ont été ignorées suite à la vérification, et finalement les fausses erreurs sont représentées par la région 2. Dans le cas idéal, c'est-à-dire une vérification parfaite de règles de dessin, les trois régions se rejoignent pour former une seule région.

La priorité de l'élimination des fausses signalisations ainsi que la signalisation des erreurs non vérifiées est indiscutable. L'expérience a montré que la proportion de fausses signalisations par rapport aux vraies erreurs peut atteindre un rapport de 100/1 voire plus dans le cas d'une vérification classique exécutée sur un dessin contenant des formes Non-Manhattan.

Le résultat direct d'un tel dysfonctionnement de l'outil de vérification de règles de dessin est le manque de confiance dans l'outil. Le concepteur microsysteme sera obligé de vérifier ses masques à la main, ce qui implique une perte énorme de temps de conception. Une mesure commune par les concepteurs est de désactiver la vérification qui génère les fausses signalisations, une telle mesure n'est pas acceptée puisqu'il laisse une partie du dessin non vérifiée.

On peut classer les sources des fausses erreurs dans un environnement de vérification de règles de dessin en deux catégories : les fausses erreurs d'origine topologique et les fausses erreurs d'origine Non-Manhattan.

5.4.1 Les fausses erreurs d'origine topologique

Dans une conception de microsystemes, il est possible d'associer différentes sortes de dispositifs sur la même couche de masque. Ces dispositifs peuvent varier entre des structures suspendues pour les parties thermiques ou mécaniques et les parties microélectroniques nécessaires pour le fonctionnement du dispositif. La même règle de dessin va varier selon la partie qui est à vérifier.

La Figure 5.16 montre une ouverture dans le substrat fréquemment utilisée pour libérer les structures suspendues en technologie AMS 0,8 μ . Du point de vue conception microsysteme en technologie AMS, la superposition des trois couches (VIA, CONT, DIFF) est nécessaire pour générer une ouverture dans le substrat, mais du point de vue conception microélectronique AMS, une telle ouverture est une violation de la règle qui impose que la couche VIA soit couverte par les couches MET1 et MET2. D'autres règles qui imposent que la couche CONT soit couverte par la couche MET1 sont également violées [13].

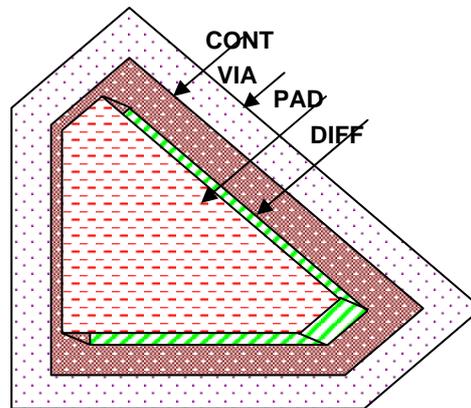


Figure 5.16. Règles microsystemes dans la technologie AMS.

Dans la technologie de SensoNor, on trouve les mêmes types de règles qui sont des dispositifs dépendants. Une règle de dessin microélectronique impose une séparation totale entre les trous d'oxyde (masque COHOL) et les dispositifs de région de NT (masque de NOSUR), alors qu'une superposition de ces deux masques est nécessaire pour se débarrasser de l'oxyde mince comme le montre la Figure 5.17 [12].

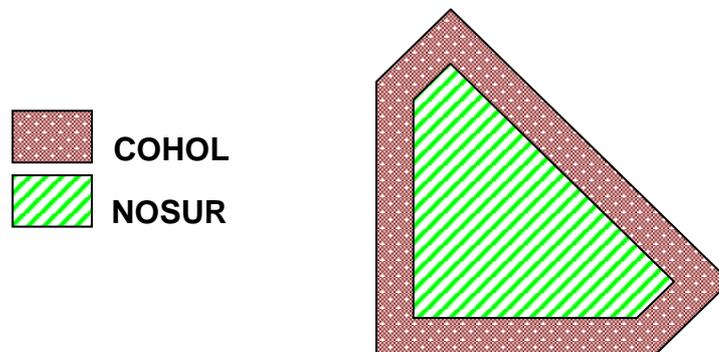


Figure 5.17. Règles microsystemes dans la technologie SensoNor.

Dans le procédé Bosch PSB technologie, le problème de la vérification des règles de dessin est plus compliqué. Dans cette technologie on trouve ce que l'on appelle des règles fonctionnelles. Une règle fonctionnelle est définie par la nature de la structure à laquelle la règle doit être appliquée. Ceci est illustré dans la Figure 5.18 [14].

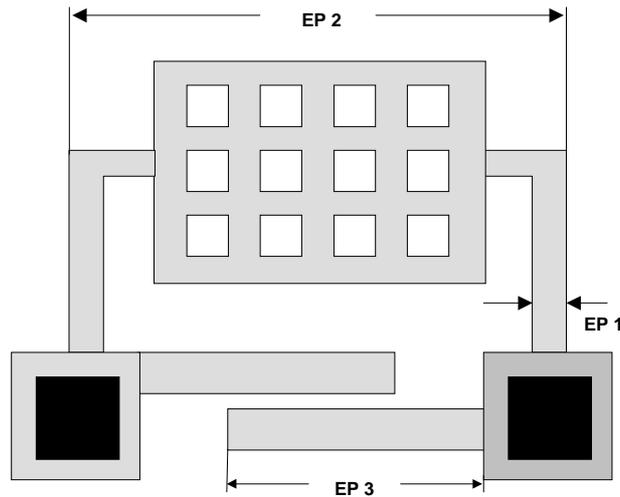


Figure 5.18. Les règles fonctionnelles dans la technologie Bosch.

Dans cette Figure trois règles de dessin sont illustrées :

La première règle (EP1) est une règle classique qui détermine la largeur minimum des structures de masque EP, la deuxième règle (EP2) fixe un autre minimum pour la largeur des structures du même masque qui sont appelées les structures libres de deux côtés. Une règle semblable (EP3) définit une troisième largeur minimum pour ce que l'on appelle des structures libres d'un seul côté. Pour les algorithmes de vérification de règles de dessin, il est quasi impossible de distinguer les trois types de structure et d'appliquer la règle appropriée à chaque structure [14].

5.4.2 Technique de résolution des problèmes topologiques

Nous avons implémenté les règles de dessin d'une manière hiérarchique. Cette méthode consiste de les trier en fonction de leur nature, puis de descendre au bas niveau avec des couches logiques, la Figure 5.19 illustre cette méthodologie :

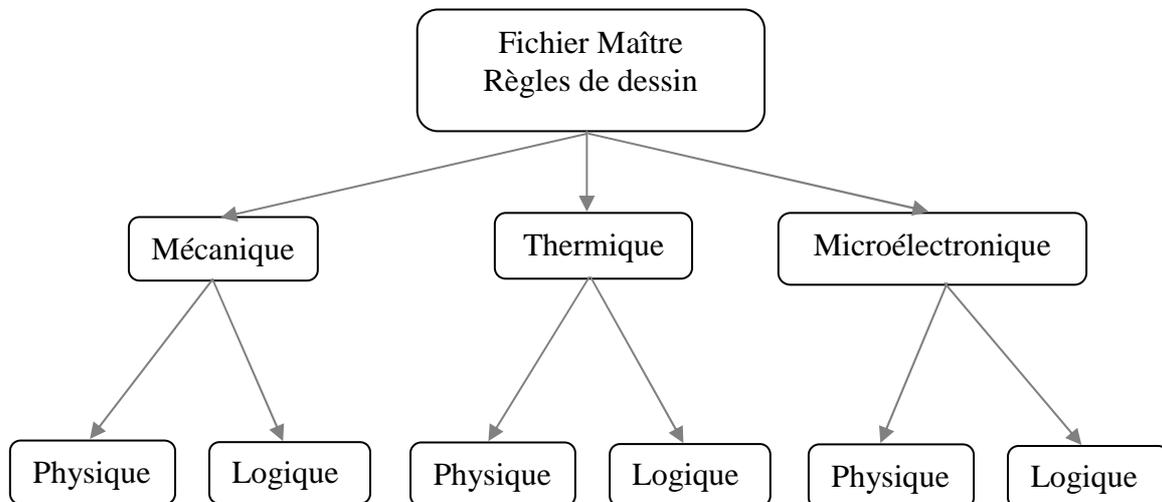


Figure 5.19. Structure hiérarchique de règles de dessin pour microsystemes.

Ainsi nous proposons à la fonderie un fichier principal qui contient des directives vers différents fichiers de règles de dessin, chacun de ces fichiers est homogène du point de vue qu'il ne contient que les règles de dessin qui concerne un domaine spécifique d'application (microélectronique, mécanique, etc.). Avec cette technique le problème des règles contradictoires est résolu.

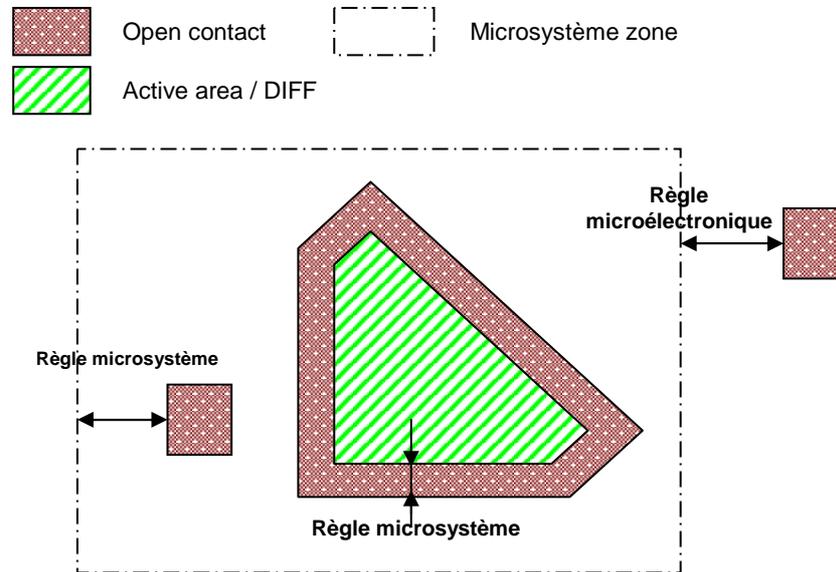


Figure 5.20. Zone microsysteme et zone microelectronique.

Au plus bas de la hiérarchie, on a des règles de dessin appliquées sur 2 types de couches : les couches physiques et les couches logiques. Les couches physiques sont les couches classiques de la technologie utilisée et les couches logiques sont les couches où l'on applique les règles fonctionnelles.

Avec cette technique le problème de la reconnaissance des formes posées par les règles fonctionnelles est résolu.

5.4.3 Les fausses erreurs d'origine Non-Manhattan

On a vu au paragraphe §.5.2 que les formes Non-Manhattan sont intensivement utilisées dans la conception des microsystemes et que ces formes Non-Manhattan sont généralement mal échantillonnées. Aussi en ce qui concerne les règles de dessin de ces formes, il y a un manque de règles de dessin qui prennent la particularité de ces formes en considération.

L'application des règles classiques de dessin à ces formes produit des milliers de fausses erreurs. On peut classer les origines de ces erreurs en trois catégories :

- Arrondissement de coordonnées des sommets de virgule flottante.
- Traduction de différents types de données (conversion d'entier vers réel ou l'inverse).
- Alignement des points sur la grille.

La plupart de ces fausses erreurs concernent principalement la largeur minimum et l'espacement minimum. La figure 5.21(a) montre le résultat d'une vérification de règles de dessin classique d'une spirale.

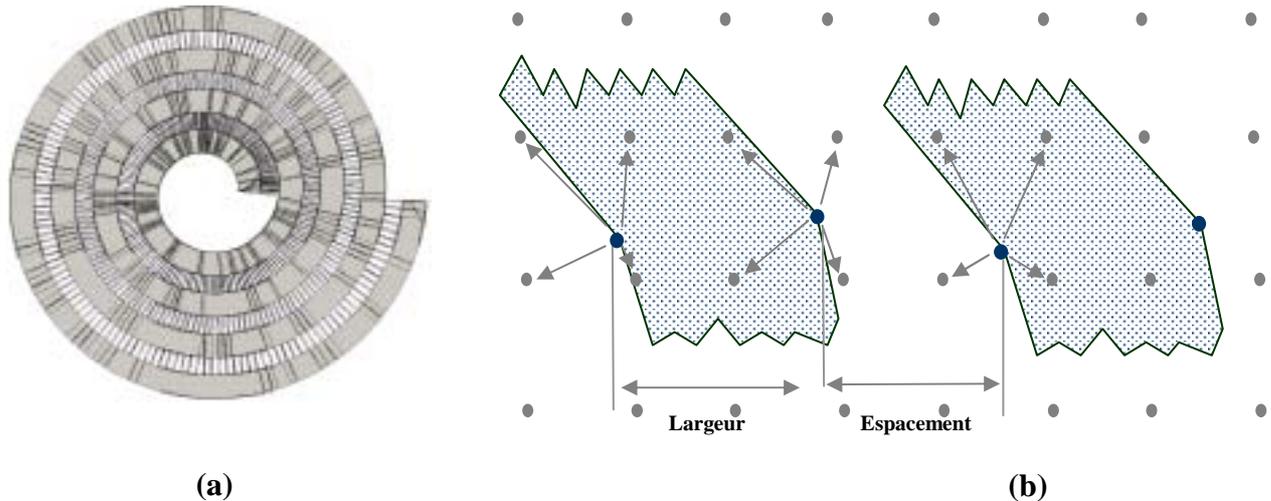


Figure 5.21. Résultats de la vérification de règles de dessin classique sur une spirale.

La Figure 5.21 (b) montre clairement comment une spirale (forme Non-Manhattan) qui respecte parfaitement les règles de dessin de largeur et espacement minimum peut devenir une source de milliers de fausses violations. On peut expliquer ce phénomène de la façon suivante :

Quand on dessine une forme Non-Manhattan avec l'éditeur de layout, elle est convertie en polygones avec un grand nombre de sommets. Normalement on a le choix d'avoir ces sommets sur la grille (*snap to grid*) ou hors grille. En outre quand on exécute l'algorithme de vérification de règles de dessin, il aligne chaque sommet à sa propre grille interne en arrondissant les coordonnées des sommets si le sommet n'est pas déjà aligné à la grille externe. Par conséquent chaque sommet subira une translation à un des quatre points voisins de la grille. L'accumulation de ces translations se traduit dans la réduction des distances entre les sommets et génère de fausses violations d'espacement et de largeur minimale.

5.4.4 Technique pour résoudre les problèmes de fausse signalisation d'origine Non-Manhattan

On définit ce que l'on appelle la tolérance de règles de dessin, c'est un paramètre dont la valeur est calculée en fonction du pas de la grille de fabrication et des paramètres d'arrondi de l'algorithme de vérification de règles de dessin.

La tolérance représente la translation maximale qu'un sommet pourrait subir en s'alignant à la grille ou en arrondissant les coordonnées de sommet, voir la Figure 5.18 (b).

Les valeurs de la tolérance pour chaque sommet sont calculées pendant une phase de pré-traitement, puis elles sont injectées dans l'algorithme de vérification des règles de dessin. Les résultats de l'introduction de la tolérance dans les règles de dessin sont illustrés dans la Figure 5.22.

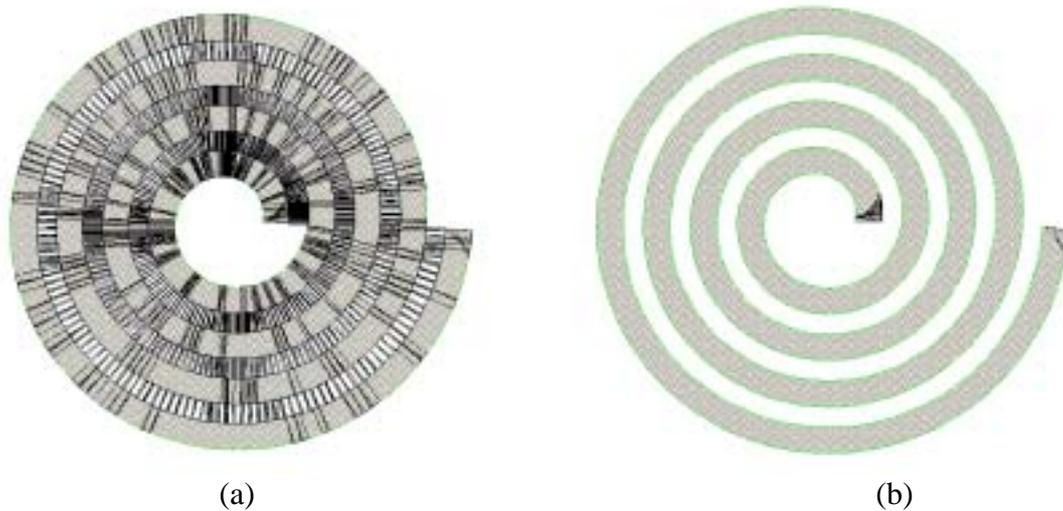


Figure 5.22. Résultats de vérification de règles de dessins : (a) exécution d'un algorithme classique, (b) exécution d'un algorithme avec la tolérance.

La Figure 5.22(a) montre les résultats de l'exécution de l'algorithme classique de vérification de règles de dessin sans la présence de la tolérance, environ 1000 violations de largeur minimum et 1500 de l'espacement minimum sont ainsi rapportées. Dans la Figure 5.22 (b) et après avoir introduit la tolérance, on voit clairement la disparition de toutes les fausses signalisations.

5.5 Conclusions

Dans ce chapitre, nous avons montré plusieurs techniques pour gérer les dessins de masques et les règles de dessin pour microsystemes. On peut regrouper ces techniques en deux catégories :

- En se contentant de ce que les outils microélectronique offrent au niveau des éditeurs de layout, c'est-à-dire des formes coniques représentées comme des polygones. On propose des techniques d'échantillonnage qui prennent en considération la particularité de ces formes. Ces techniques sont utilisées dans la génération des formes coniques et dans les algorithmes de vérification de règles de dessin.
- Soit en adoptant une nouvelle génération d'éditeurs de layout, dans de tels éditeurs de layout les formes géométriques sont représentées par leurs équations analytiques et non pas par des polygones à grands nombres de sommets. Cette approche est très efficace pour réduire le temps de vérification des règles de dessin.

Références du Chapitre 5

- [1] T. A. Anderson, C. E. Kim, "Representation of digital straight lines and their preimages", *7th ICPR, IEEE Comp. Soc. Press*, pp.501-504, Montréal, June 1984.
- [2] C. Arcelli, "A condition for digital points removal", *Signal Processing, 1*, pp. 283-285, 1979.
- [3] C. Arcelli, G. Sanniti di Baja, "Shape Splitting using maximal neighborhood", *6th ICPR, IEEE Comp. Soc. Press*, pp. 1106-1108, Munich, October 1982.
- [4] R. Bellman, "On the approximation of curves by line segments using dynamic processing", *Comm. Of ACM, 4*, pp.284, 1964.
- [5] Berenstein C.A., Lavine D., "On the number of Digital straight line segments", *IEEE trans. on PAMI, 10(6)*, pp. 880-887, November 1988.
- [6] A. Bogomolny, "Digital geometry may not be discrete", *CVGIP, 43*, pp.205-220,1988
- [7] Brons R., "Linguistic methods for the description of a straight line on a grid", *CGIP, 3*, pp.48-62, 1974.
- [8] L. Dorst, W. M. Smeulders, "Discrete Representation of straight lines ", *IEEE trans. on PAMI, 6(4)*, pp. 450-463, 1984.
- [9] <http://www.tanner.com/>
- [10] <http://www.mentor.com/>
- [11] <http://www.cadence.com/>
- [12] G. Bredholt, C. Merveille, "SensoNor Foundry Process Design Handbook", *SensoNor asa*, 1999.
- [13] D. Veychard & al., "Design Rules for Front Side Bulk Micromachined Microstructures", *TIMA-CMP*, 1998.
- [14] M. Fehrenbach & al., "Micromachining Foundry Design Rules", *BOSCH*, 1999.
- [15] G. S. Taylor and John K. Ousterhout, "Magic's Incremental Design-Rule Checker", *Proc. 21st Design Automation Conference*, June, 1984, pp. 11.2.
- [16] Ann R. Lanfri, "PHLED45: An Enhanced Version of CEASAR Supporting 45° Geometries", *Proc. 21st Design Automation Conference*, June, 1984, pp. 34.1.
- [17] Edward J. McGrath & al., "Design Integrity and Immunity Checking", *Proc. 17th Design Automation Conference*, June, 1980. Page. 263.

- [18] A. Tsukisoe & al., "MACH: A High-Hitting Pattern Checker for VLSI Mask Data", *20th Design Automation Conference*, June, 1983, pp.461.
- [19] Z. Juneidi., K. Torki, R. Hamza, "Design Rules for Non Manhattan Shapes", *SPIE Micromachining and Microfabrication*, Santa Clara, CA, USA, September 2000.

6 Techniques de synthèse et d'optimisation de microsystemes

6.1 Introduction

Avec le développement rapide de microsystemes, il y a une demande pour un certain type d'outils de synthèse. Ces outils doivent permettre la traduction d'une description de haut niveau d'abstraction dans une conception valide de microsysteme. Un tel outil de synthèse permet aux concepteurs d'explorer un espace de conception défini par plusieurs types de contraintes et de concevoir des hiérarchies de composants microsystemes à partir des cellules de base.

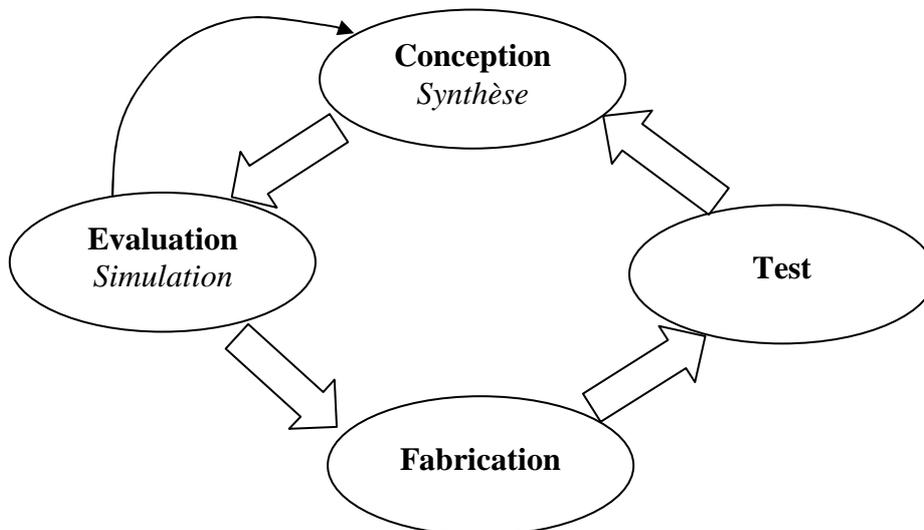


Figure 6.1. Schéma simplifié de cycle de conception microsysteme.

Dans le cycle de conception de microsystemes illustré dans la Figure 6.1, la phase finale avant la fabrication est l'étape de dessin des masques. Il s'agit en effet de réaliser tous les motifs des masques qui seront utilisés lors de la fabrication du microsysteme. Pour y procéder, le concepteur a besoin de prendre des décisions de quel type de topologie il doit disposer, de faire de multiples itérations entre plusieurs spécifications qui peuvent satisfaire aux différentes performances. Ceci afin d'assigner les valeurs qui décrivent une conception valide à un ensemble de paramètres. Le grand nombre de variables et la non linéarité des équations qui les relient rend difficile la recherche d'un tel ensemble de valeurs [1] [3].

L'intérêt d'automatiser cette phase de la conception est outre un gain de temps considérable lors du dessin du masque, de permettre aussi la suppression des multiples itérations qui étaient jusqu'alors nécessaires avant d'obtenir un layout correct. Tout ceci entraînant, bien sûr, un énorme gain de productivité [2].

La plus grande partie des outils de layout automatique existants concerne les circuits intégrés numériques. L'électronique numérique est en effet le seul domaine pour lequel la recherche a débouché sur des logiciels utilisés dans l'industrie. Le développement de ces logiciels a été facilité par l'introduction d'une certaine standardisation dans les circuits logiques. Cette standardisation consiste à réutiliser au maximum, au sein d'un même circuit, un certain nombre de structures prédéfinies. Elle permet donc de simplifier la tâche du concepteur en limitant ses degrés de liberté. Cette simplification se fait au détriment de la taille du circuit et de ses performances. Un compromis doit donc être trouvé entre la diminution du temps de conception du circuit et l'augmentation de sa taille. Différents degrés de standardisation sont donc possibles, chacun correspondant à un style de layout différent [1] [2] [3].

En ce qui concerne les microsystemes (comme dans le domaine de l'analogique), une telle standardisation n'est pas envisageable. En effet, les tailles des composants utilisés au sein d'un même microsysteme varient beaucoup plus et il est quasiment impossible de réaliser des bibliothèques de blocs fonctionnels : en effet, même s'ils réalisent la même fonctionnalité, ceux-ci doivent être redimensionnés à chaque fois, voire même complètement reconçus, pour satisfaire aux spécifications du système global. Il est donc particulièrement difficile de trouver des simplifications dans la manière de procéder pour réaliser le dessin des masques d'un microsysteme [5] [7].

Celui-ci comporte en général trois étapes : la création des layouts de toutes les cellules élémentaires du microsysteme, le placement de ces cellules les unes par rapport aux autres de manière optimisée, et enfin la réalisation des interconnexions. Cette méthodologie serait proche de certains types de dessin des masques en numérique si les phases d'optimisation ne faisaient pas intervenir les contraintes indispensables au bon fonctionnement final du microsysteme. Le principal problème devient alors non pas d'arriver à connecter entre elles toutes les cellules du microsysteme (il n'y en a heureusement pas beaucoup), mais bien d'assurer que celui-ci fonctionnera correctement à la fin, même si cela doit entraîner une certaine perte de place. Nous sommes passés d'un problème d'ordre plutôt quantitatif à un problème plus qualitatif, qu'il est d'autant plus difficile de traiter de manière informatique. C'est pourquoi, à notre connaissance, il n'existe actuellement que peu de systèmes d'automatisation du layout en microsysteme [4] [8].

Dans ce chapitre, nous allons donc présenter une méthodologie de synthèse et d'optimisation pour les microsystemes, cette méthodologie doit conduire à la génération automatique du layout des microsystemes à partir d'un modèle comportemental.

6.2 Méthodologie de synthèse pour microsystème

L'objectif d'un outil de synthèse de microsystème est de traduire une description à haut niveau dans une conception physique de microsystème. Pour cela le système est décrit par un langage hybride qui combine la description comportementale et certaines directives que l'on ajoute à la syntaxe du langage HDL-A. Ces directives conduisent un «*parseur*» pour extraire ce que l'on appelle des paramètres de conception. Les contraintes de conception sont implémentées dans une base de données ainsi que les paramètres de la technologie utilisée. La figure suivante illustre un tel environnement [6] :

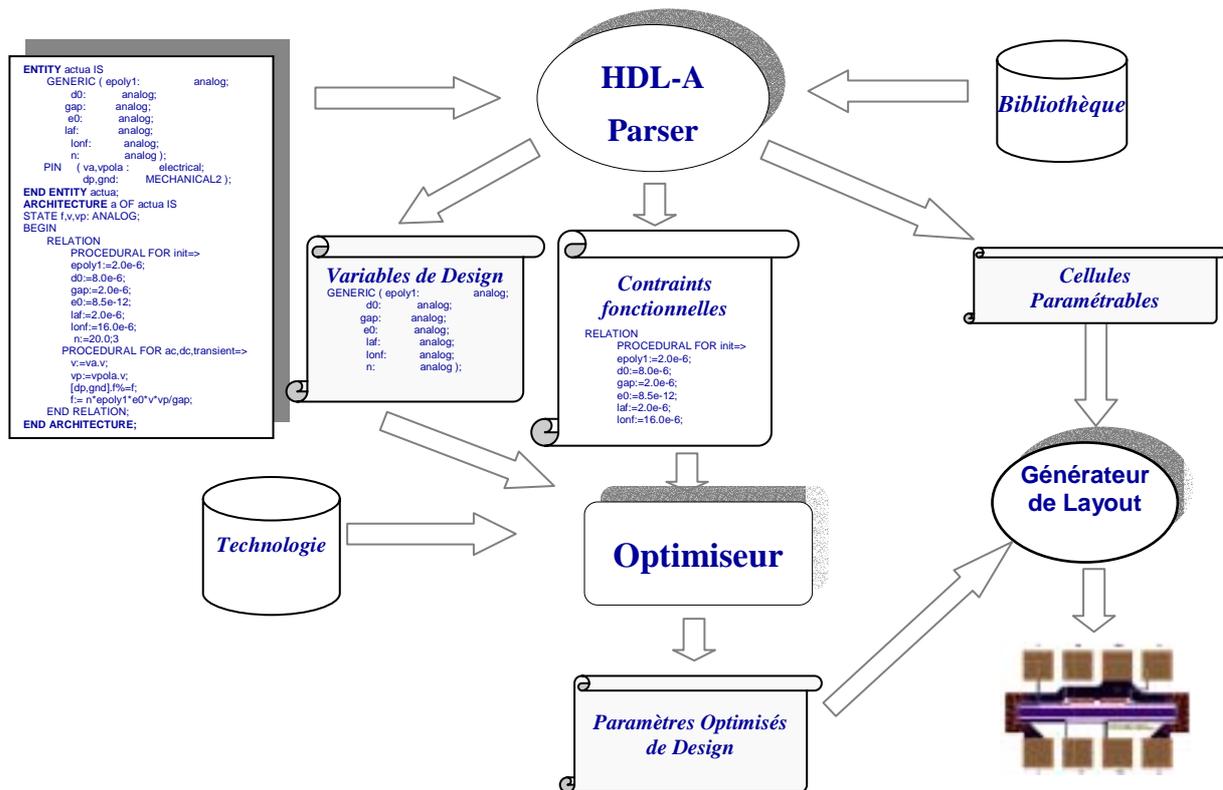


Figure 6.2. Environnement de synthèse pour microsystème.

On va détailler dans les sections suivantes les différents modules de cet outil de synthèse. Pour illustrer cette méthodologie, on va utiliser comme application le convertisseur électrothermique détaillé au § 4.3.2.2.

Il y a deux catégories de paramètres qui déterminent le fonctionnement du convertisseur électrothermique :

- 1- Les paramètres technologiques liés à la technologie utilisée, comme la conductivité des matériaux, la résistance, etc.
- 2- Les paramètres de conception : ce sont les paramètres liés à la forme du convertisseur, comme la longueur de la poutre, sa largeur, le nombre de thermocouples, etc.

La Figure 6.3 montre les paramètres de conception sur une structure de convertisseur électrothermique.

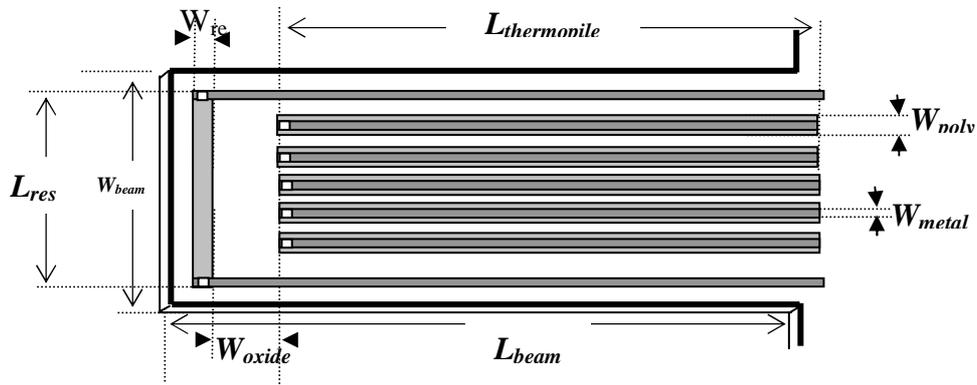


Figure 6.3. Paramètres de conception sur une structure de convertisseur électrothermique.

Ces paramètres sont stockés dans un fichier « générique », on fait appel à ce fichier dans toute description hybride pour les composantes élémentaires [6].

Dans le notre cas, le fichier « générique » contient les paramètres suivants :

```

Generic
z:                integer;    --nbr de thermocouples--
alpha1, alpha2:  analog;    --coef seebeck--
tcrcpoly:        analog;    --coef de variation de resistance
                    avec la temperature--
Rsqpoly:         analog;    --resistance carre de poly--
Rsqmet:          analog;    --resistance carre du metal--
gmet:            analog;    --conduction thermique du metal--
gpoly:           analog;
cpoly:           analog;    --capacite thermique du poly--
hair:           analog;    --convection thermique de l'air--
emit:            analog;    --emittance--
sb:              analog;    --constante de stephan boltzmann--
gox:             analog;    --conductance thermique de l oxide--
gpass:          analog;
cox:             analog;    --capacite thermique de l oxide--
T0:              analog;    --temperature ambiante--
elocos:         analog;    --epaisseurs--
epoly1:         analog;
epoly2:         analog;
eox1:           analog;
eox2:           analog;
eox3:           analog;
eox4:           analog;
emet1:          analog;
emet2:          analog;
lonptr:         analog;    --longueurs, largeurs--
larptr:         analog;
lontp:         analog;
lonox:         analog;
larpoly1:       analog;
larpoly2:       analog;
larmet:         analog;
lonres:         analog;
larres:         analog;
aox:           analog;
atp:           analog;
    
```

Si on regarde le modèle HDL-A du convertisseur électrothermique en détail, on peut remarquer qu'il est composé de plusieurs composants élémentaires qui sont des briques de base comme il est montré à la Figure 6.4.

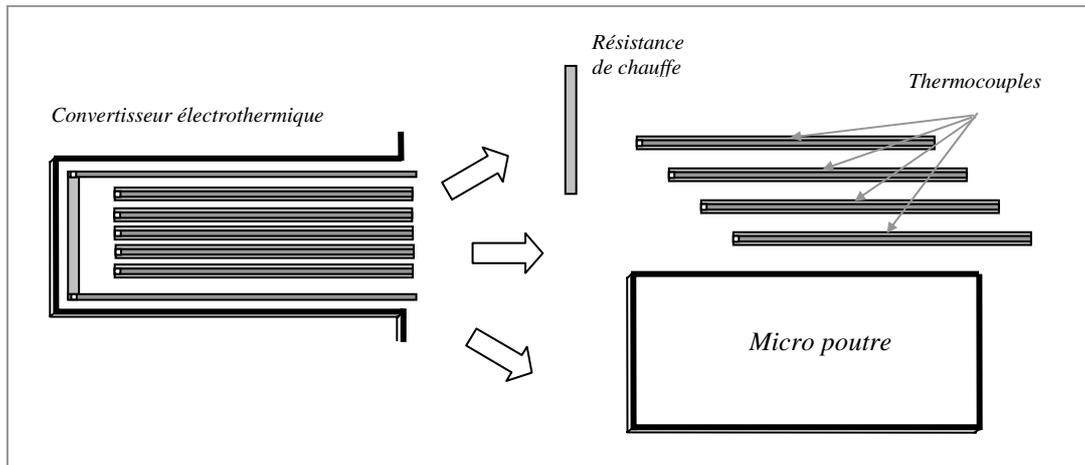


Figure 6.4. Décomposition d'un convertisseur électrothermique dans des éléments de base.

- **Thermocouple :**

Le thermocouple est une superposition de deux métaux différents comme illustré dans la Figure 6.5.

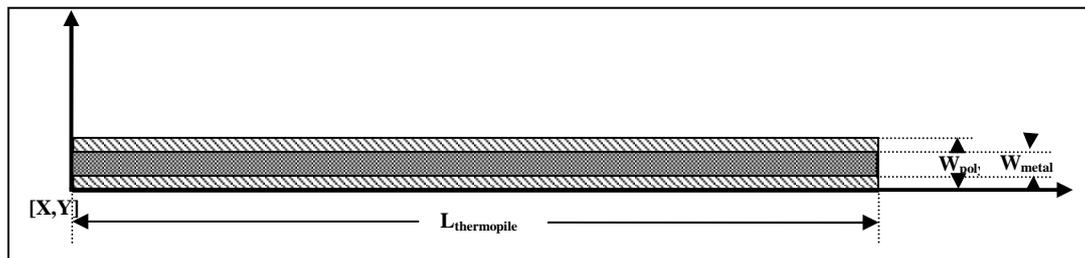


Figure 6.5. Structure d'un thermocouple

Du point de vue comportemental, un thermocouple est une boîte noire avec une entrée et une sortie :

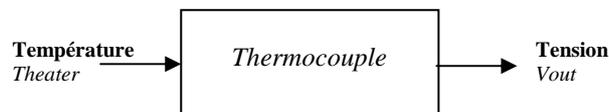


Figure 6.6. Entrée/sortie d'un thermocouple.

On propose la description suivante à ce composant :

```

component thermocouple [X,Y,Length_thermopile,Wedth_metal1, Wedth_metal2, metal1,metal2]
pin(
    theater,t0:          electrical;    --sorties--
    voutp,voutn:        electrical;    --entrees--
)

include "generic";
state rthox:          analog;
state cthox:          analog;
state Sox:            analog;
Geometric
    Begin
        Rectangle [X,Y, Length_thermopile, Wedth_metal1, metal1];
        Rectangle [X,Y+ 0.5 Wedth_metal1 -0.5 Wedth_metal2, Length_thermopile,
        Wedth_metal2, metal2];
    End
Behavioral
    Begin
        cthox:=cox*Wedth_metal2*(elocos+eox1+eox2+eox3+eox4)*length_metal2;
        rthox:=gox*Length_metal2*(elocos+eox1+eox2+eox3)/Lengt_metal1*gmet*
        width_metal1*emet1/width_metal2*2.0+gpass*width_metal2*(eox4)/length_metal1;
        v0:=(tp-T0)*(alpha1-alpha2)+i0*rtp;
        rtp:=rsqpoly*(lontp/Width_metal1+lontp/Width_metal2)*(1.0+tcrcpoly*(tp-T0));
        procedural for dc =>
            vin:=[vinp,vinn].v
            [voutp,voutn].v%=v0;
            i0:=voutp.i;
            Tth.v%=theater;
            Toxi.v%=tox;
            v0:=(tp-T0)*(alpha1-alpha2)+i0*rtp;
            rtp:=rsqpoly*(lontp/Width_metal1+lontp/Width_metal2)*(1.0+tcrcpoly*(tp
            -T0));
        equation(theater,tox,tp) for dc =>
            aox*rthox*(theater-tox)==(1.0-aox)*(rthox)*(tox-tp1)+(tox-
            T0)*Sox*hair+4.0*emit*Sox*sb*T0*T0*T0*(tox-T0);
            (1.0-aox)*(rthox)*(tox-tp)==atp*(rthtp)*(tox-tp);
        procedural for tranient =>
            vin:=[vinp,vinn].v
            [voutp,voutn].v%=v0;
            i0:=voutp.i;
            Tth.v%=theater;
            Toxi.v%=tox;
            v0:=(tp-T0)*(alpha1-alpha2)+i0*rtp;
            rtp:=rsqpoly*(lontp/Width_metal1+lontp/Width_metal2)*(1.0+tcrcpoly*(tp
            -T0));
        equation(theater,tox,tp) for tranient =>

            aox*rthox*(theater-tox)==cthox*tox'ddt+(1.0-aox)*(rthox)*(tox-
            tp)+(tox-T0)*Sox*hair+4.0*emit*Sox*sb*T0*T0*T0*(tox-T0);
            (1.0-aox)*(rthox)*(tox-tp)==atp*(rthtp)*(tox-tp);

    end
    
```

- **Résistance de chauffe :**

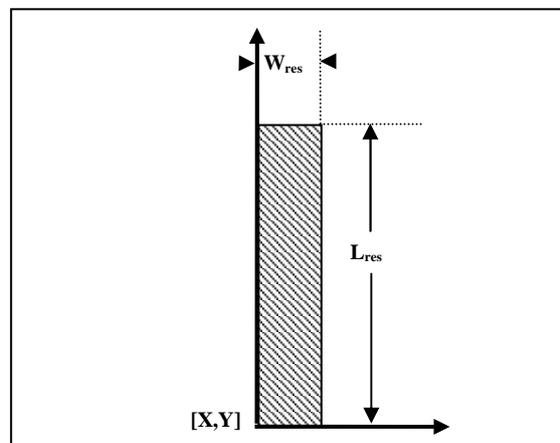


Figure 6.7. Résistance de chauffe.

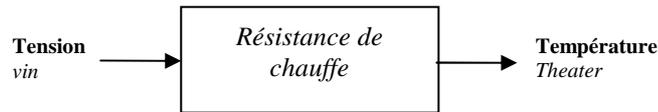


Figure 6.8. Entrée/sortie d'un thermocouple.

Et la description correspondante :

```

component Resistance [X,Y,Wedth_Res,Length_Res,Metal]
pin(
    vinp,vinn:          electrical;    --entrees--
    theater,t0:         electrical;    --sorties--
)
include "generic";
state cthres:         analog;
state i0:             analog;
state V0:             analog;

Geometric
    Rectangle [X,Y,Wedth_Res,Length_Res,Metal];
Behavioral
begin
    cthres:=cpoly*epoly1*Length_Res*Wedth_Res+cox*(elocos+eox1+eox2+eox3
    +eox4)* Length_Res*2.0*Wedth_Res+cox;
    procedural for dc =>
        vin:=[vinp,vinn].v;
        Tth.v%=theater;
    equation(theater) for dc =>
        vin*vin/(rsqpoly*Width_Res/lelgth_Res*(1.0+tcrcpoly*(theater-
        T0)))=aox*(rthox)*(theater-tox);
end
    
```

• La Micro poutre:

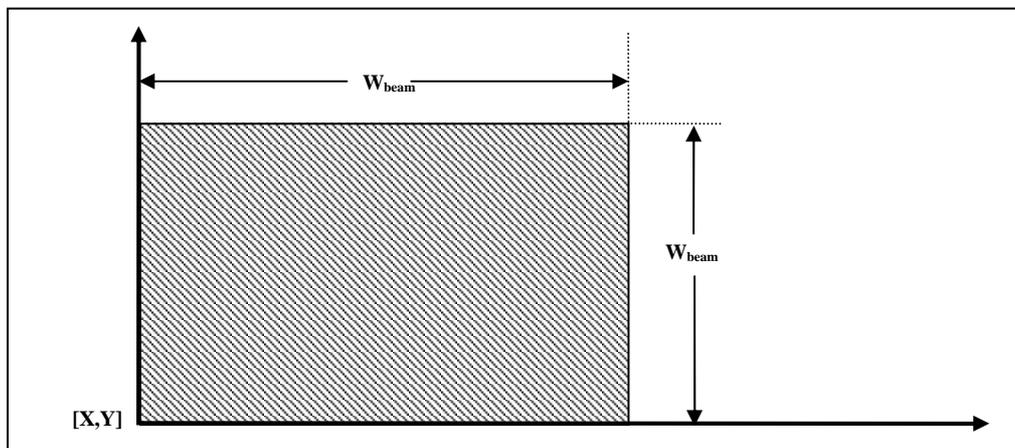


Figure 6.9. La poutre

```

component Beam [X,Y,Length_Beam,Wedth_Beam,Metal]
Begin
    Rectangle [X,Y,Length_Beam,Wedth_Beam,Metal];
end
    
```

Avec les composants élémentaires développés, l'architecture d'un convertisseur électrothermique peut avoir la forme illustrée dans la figure suivante :

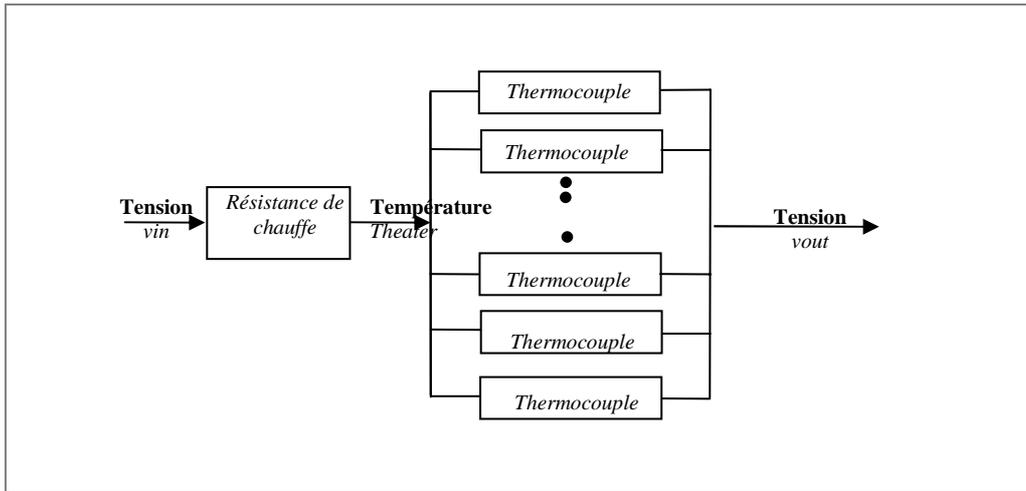


Figure 6.10. L'architecture d'un convertisseur électrothermique.

Finalement, une description hybride d'un convertisseur électrothermique a la forme suivante :

```

Component convertir [x,y,z]
pin(   vinp,vinn:           electrical;   --entrees--
      voutp,voutn:        electrical;   --sorties--
      Tth,Toxi,T:         electrical    --temperatures--
    );

Integer i;
Float x_Res_Edge, y_Res_Edge;
Begin
  Beam [x,y,Length_Beam,Wedth_Beam,Metal];
  Resistance [X,Y,Wedth_Res,Length_Res,Metal];
  Interface (Resistance,thermocouple).
  For i=2 to i=z
  Begin
    thermocouple [X,Y,Length_thermopile,Wedth_metal1,
                  Wedth_metal2, metal1,metal2]
    Interface(thermocouple,thermocouple).
  end
end
end
    
```

6.3 Technique d'optimisation

L'optimisation est un sujet central dans toutes les démarches de conception. Tout concepteur a normalement pour but de créer des composants optimaux. Optimal est à prendre dans le sens que le composant ait des performances aussi proches que possible de celles spécifiées à priori ou que ce composant tire le meilleur parti possible de la technologie disponible pour un effort donné. L'optimisation peut être utilisée pour plusieurs catégories d'objectifs :

- La recherche du meilleur composant possible.
- La recherche d'un composant amélioré.
- La recherche d'un composant qui marche.

Les problèmes d'optimisation sont constitués de trois ingrédients :

1. La fonction objective ou coût que l'on cherche à minimiser ou maximiser. Par exemple, dans un procédé de fabrication, nous cherchons un maximum de profit et un minimum de coût. Autre exemple, dans les applications d'ajustement «*fitting*» des données expérimentales par une courbe analytique, nous tâcherons de minimiser la déviation totale séparant les points expérimentaux de leurs modèles mathématiques. En pratique, la plupart des problèmes d'optimisation ont une seule fonction objective. Cependant deux exceptions peuvent se présenter :
 - Il n'y pas de fonction coût. C'est le cas, par exemple, de la conception physique (layout) d'un circuit intégré. Le but est de trouver un ensemble de variables qui satisfassent les contraintes du modèle sans qu'il n'y ait d'autres buts à atteindre. Cette catégorie de problèmes est généralement désignée par les problèmes de faisabilité.
 - Fonction objective multiple : souvent, l'utilisateur souhaite minimiser (ou maximiser) plusieurs caractéristiques à la fois. C'est le cas d'un circuit où l'on voudrait un profil de gain et une figure de bruit par exemple. C'est aussi le cas d'une inductance où l'on exige à la fois un profil d'inductance propre et un profil de facteur de qualité. Dans la pratique, les problèmes avec plusieurs fonctions objectives sont reformulés comme les problèmes à une seule fonction objective soit en formant une combinaison pondérée des différentes fonctions objectives, soit en transformant certaines fonctions sous la forme de contraintes sur les variables.
2. Variables ou paramètres d'optimisation qui affectent la fonction objective : dans l'exemple de procédé de fabrication, les variables peuvent être le nombre de ressources utilisées ou le temps consacré à chaque activité. Dans l'application d'ajustement des données les variables sont le coefficient du modèle analytique, etc. A noter que la présence des variables dans un problème d'optimisation est essentiel car nous ne pouvons rien définir en leur absence.
3. Un ensemble de contraintes autorisant les variables à prendre un certain nombre de valeurs tout en excluant d'autres. Le temps doit être positif dans le cas du procédé de fabrication. Les coefficients du modèle analytique peuvent être liés par des relations imposées par des considérations physiques ou géométriques, etc. Cet ensemble de contraintes peut aussi être vide et pour un problème donné les variables peuvent être complètement indépendantes les unes des autres. Il est à noter que la complexité algorithmique d'un problème donné est très fortement liée à la présence ou non de ces contraintes et à la nature des relations de contraintes entre les variables. D'ailleurs, c'est ce facteur que l'on va utiliser lors de la classification des méthodes d'optimisation comme nous le verrons plus tard.

6.3.1 Processus d'optimisation

Un processus d'optimisation peut schématiquement être décomposé en quatre étapes (Figure 6.11) :

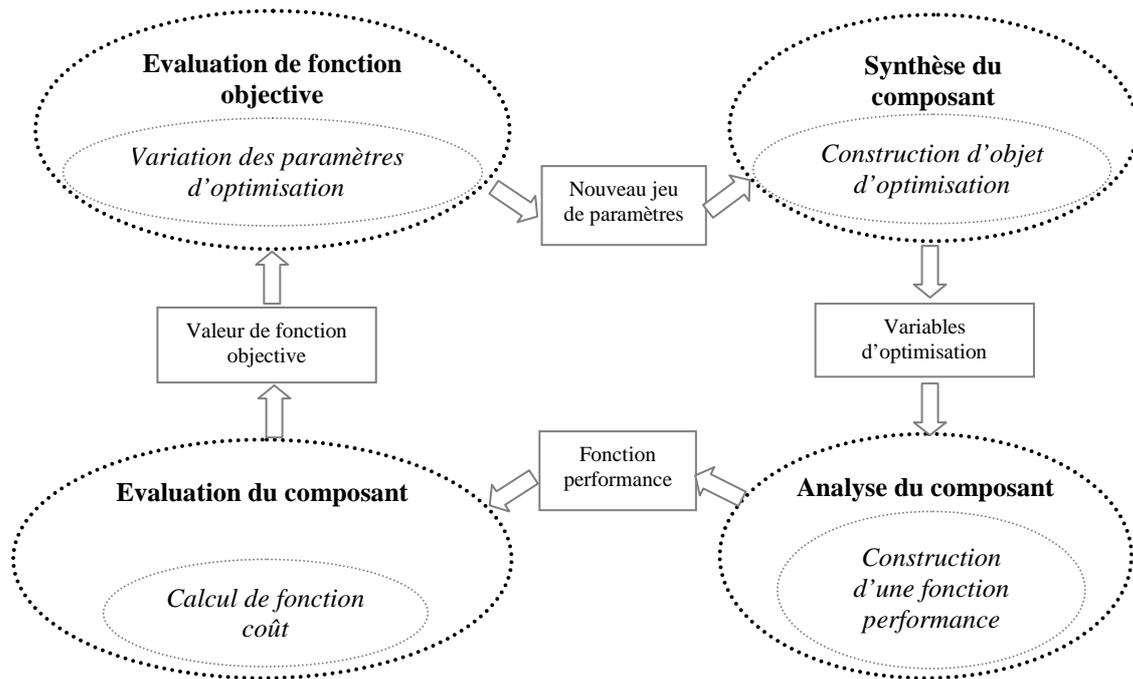


Figure 6.11. *Processus d'optimisation.*

1. Synthèse d'un nouveau composant : cette étape est consacrée à la construction d'un nouvel objet d'optimisation à partir des paramètres de construction (variables).
2. Analyse du composant : il s'agit d'extraire du composant synthétisé à l'étape 1 les informations primaires nécessaires à la construction de la fonction coût.
3. Evaluation du composant : les informations primaires extraites à l'étape 2 doivent être transformées en une unique valeur numérique représentant la fonction objective. Cette valeur numérique servira de base à la comparaison du composant courant avec les autres.
4. Variation des paramètres d'optimisation : la valeur de la fonction objective de l'étape précédente donnera la base à partir de laquelle un nouvel ensemble de variables est calculé et un rebouclage sur l'étape 1 est effectué.

Il est important de remarquer que toutes ces étapes doivent être initiées soit manuellement, soit par des procédés mathématiques ou empiriques. Il va de soi que le succès du processus d'optimisation est en partie lié aux bons choix initiaux [9].

A remarquer aussi que du point de vue mathématique, les étapes les plus cruciales sont les étapes 3 et 4 car toute l'algorithmique nécessaire à l'optimisation y est implémentée. Il est aussi à noter que les étapes 1 et 2 supposent l'existence d'un outil (simulateur) permettant l'obtention des informations primaires sur le composant connaissant les valeurs des variables. Nous supposons cette étape assurée par des simulateurs externes existant et l'on ignore sa complexité mathématique dans le processus d'optimisation [9].

6.4 Optimisation de microsysteme : optimisation d'un convertisseur électrothermique

6.4.1 Introduction

On peut formuler un problème d'optimisation sous contrainte dans \mathfrak{R}^n par :

$$(P) \begin{cases} \text{Minimiser } f(x) \\ \text{sous les Contraintes :} \\ g_i(x) \leq 0 \quad i = 1..m \\ x \in S \subset \mathfrak{R}^n \end{cases}$$

Dans le cas d'un convertisseur électrothermique, on s'intéresse à optimiser deux grandeurs essentielles :

1. La quantité de silicium nécessaire pour fabriquer le composant, ce qui se traduit par la minimisation de la surface du composant.
2. La réponse thermique du convertisseur : ce qui signifie la maximisation de son amplitude R et de sa constante de temps τ , voir la Figure 6.12.

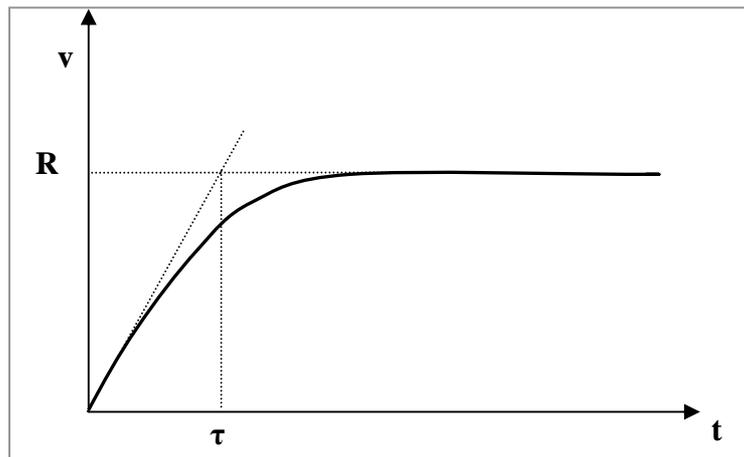


Figure 6.12. Paramètres R et τ de la réponse du convertisseur électrothermique.

On peut avoir plusieurs fonctions objectives à minimiser-maximiser :

- Maximiser la fonction τ « constante de temps ».
- Maximiser la fonction R « réponse ».
- Minimiser la fonction S « surface de composant ».

Les limites de ces fonctions sont données par le tableau suivant :

Limites des fonctions objectives				
Fonction	Description	Min	Max	Unité
S	Surface	3000	15000	μm^2
$\tau = RC$	Constante de temps	0	50	10^{-3}s
R	Réponse ETC	0	100	10^{-2}V

Tableau 6.1. Limites de fonctions objectives pour un convertisseur électrothermique fabriqué en technologie AMS 0,6 μm .

R et τ sont des fonctions multi-variables. Ces variables sont les paramètres qui définissent la géométrie du composant ; paramètres de conception. Le Tableau 6.2 montre les paramètres de conception avec leurs minima et maxima :

Paramètres de Conception			
Var.	Description	Min	Max
L_{beam}	Longueur de la poutre	100e-6	500e-6
W_{beam}	Largeur de la poutre	30e-6	300e-6
L_{res}	Longueur de la résistance	10e-6	300e-6
W_{res}	Largeur de la résistance	0.8e-6	10e-6
W_{poly}	Largeur de Poly	0.9e-6	10e-6
W_{metal}	Largeur de Metal	0.8e-6	10e-6
L_{thp}	Longueur du thermocouple	20e-6	500e-6
W_{oxide}	Largeur de Oxide	8e-6	500e-6
N	Nombre de thermocouple	2	30
$\frac{L_{beam}}{W_{beam}}$	Facteur de forme	1	5

Tableau 6.2. Minima et maxima géométriques des différents paramètres de conception.

En général, les minima et les maxima de ces paramètres sont fixés par la technologie utilisée. Dans notre cas, c'est la technologie AMS 0,6 μ m qui est utilisée pour la fabrication de ce convertisseur.

Pour ce qui concerne la fonction de surface S , sa valeur est donnée par la multiplication de longueur et de la largeur de la micro poutre : $S=L_{beam}W_{beam}$.

Le problème d'optimisation du convertisseur électrothermique n'est pas un problème d'optimisation avec une seule fonction objective. Il est inutile de minimiser la surface du convertisseur (un problème d'optimisation à deux variables (L_{beam}, W_{beam}) et à une seule contrainte "Facteur de forme") si cela va détériorer la performance du composant.

On se retrouve dans le cas d'une fonction objective multiple, c'est-à-dire avoir plusieurs combinaisons de fonctions objectives, par exemple « maximiser τ et minimiser S », ou « maximiser R et minimiser S », etc. Dans la pratique, les problèmes avec plusieurs fonctions objectives sont reformulés comme les problèmes à une seule fonction objective, soit en formant une combinaison pondérée des différentes fonctions objectives, soit en transformant certaines des fonctions objectives sous la forme de contraintes sur les variables. Dans le cas d'une optimisation multiple des trois fonctions objectives (τ , R , S) d'un convertisseur, on transforme deux de ces trois fonctions objectives en contraintes sur des variables. Par exemple, si notre problème est d'avoir $\tau=40$, $S=300$ et $R=90$ on garde une de ces fonctions comme fonction objective (τ) et on transforme les deux autres en deux contraintes supplémentaires $300 \leq S \leq 305$ et $90 \leq R \leq 95$.

Dans la suite, nous allons traiter le problème d'optimisation du convertisseur électrothermique dans le cas le plus général, c'est-à-dire avoir une fonction objective multiple de trois grandeurs. Ce problème est formulé de la façon suivante :

$$\left. \begin{array}{l}
 \text{Minimiser } S = l_{beam} \cdot w_{beam} \\
 \text{Maximiser } R = f(l_{beam}, w_{beam}, l_{res}, w_{res}, l_{thp}, w_{oxide}, w_{metal}, w_{poly}, N) \\
 \text{Maximiser } \tau = f(l_{beam}, w_{beam}, l_{res}, w_{res}, l_{thp}, w_{oxide}, w_{metal}, w_{poly}, N)
 \end{array} \right\} \text{optimiser le convertisseur}$$

$$\left. \begin{array}{l}
 100e-6 \leq l_{beam} \leq 500e-6 \\
 30e-6 \leq w_{beam} \leq 300e-6 \\
 10e-6 \leq l_{res} \leq 300e-6 \\
 0.8e-6 \leq w_{res} \leq 10e-6 \\
 20e-6 \leq l_{thp} \leq 500e-6 \\
 8e-6 \leq l_{oxide} \leq 500e-6 \\
 0.9e-6 \leq w_{poly} \leq 10e-6 \\
 0.8e-6 \leq w_{metal} \leq 10e-6 \\
 1 \leq l_{beam} / w_{beam} \leq 5 \\
 2 \leq N \leq 30
 \end{array} \right\} \text{sous les contraintes :}$$

6.4.2 Expérimentation et résultats

Le processus d'optimisation du convertisseur électrothermique s'effectue selon les étapes suivantes :

- **Analyse de paramètres pour définir leur nature (discrète ou continue) :** les algorithmes d'optimisation varient selon les paramètres à optimiser. Un des problèmes les plus difficiles à résoudre dans le domaine de l'optimisation est la combinaison entre des paramètres discrets et des paramètres continus. Il y a deux techniques pour surmonter ce problème :
 - La première technique consiste à effectuer des transformations des variables discrètes en variables continues, ce qui permet d'utiliser les méthodes d'optimisation aux variables continues, une fois que l'on a les valeurs continues des paramètres, on procède aux transformations inverses (continues vers discrètes). Dans la plupart des cas, ces transformations sont des arrondis simples de réels en entiers. Cette méthode est employée quand le nombre des variables est important (plus de 10 variables). L'inconvénient de cette méthode réside dans le fait que les arrondis des variables conduisent souvent à une divergence de l'algorithme d'optimisation [11].
 - La deuxième technique concerne des cas particuliers où le nombre des variables discrètes est limité (1 seul variable ou deux). C'est le cas du convertisseur électrothermique où l'on a une seule variable discrète : N, le nombre des thermocouples. Dans ce cas particulier, on isole les paramètres continus, on procède à l'optimisation de ces paramètres en rebouclant sur les variables discrètes (une seule boucle dans le cas d'un convertisseur électrothermique), pour chaque

itération (N=2 à N=30) on choisit le jeu de paramètres qui donne la performance la plus proche de la performance désirée. A la fin du cycle on élit la performance la plus adéquate [11].

- **Choix des algorithmes d'optimisation :** l'annexe D donne une idée sur la variété de problèmes de l'optimisation et les différentes méthodes développées pour les résoudre. Il est clair que dans le cas d'optimisation d'un convertisseur électrothermique, il s'agit d'un problème d'optimisation non linéaire à variables continues, si l'on isole N comme expliqué dans le paragraphe précédent. Pour résoudre ce problème, on a recours aux deux méthodes d'optimisation non linéaires à variables continues sans contraintes avec un variant « fonction de pénalité » pour les transformer en méthodes d'optimisation avec contraintes [9] [10] [11] :
 - **Le gradient de la plus forte pente descendante :** cette méthode est dite newtonienne, il s'agit d'abord de calculer le gradient de la fonction à optimiser (τ , R ou S) qui est le vecteur de dérivées partielles premières (vecteur de 8 composants) de fonction par rapport aux variables d'optimisation (l_{beam} , w_{beam} , l_{res} , w_{res} , l_{thp} , l_{oxide} , w_{poly} , w_{metale}), puis de calculer la matrice hessienne de la fonction qui est la matrice des dérivées partielles secondes (une matrice B [8x8]). Avec les valeurs de gradient on définit la direction de convergence, et avec la matrice hessienne on optimise la vitesse de convergence.
 - **La méthode de Broyden-Fletcher-Goldfarb-Shanno, BFGS :** cette méthode est dite quasi newtonienne, comme dans les méthodes newtoniennes, la première étape consiste à de calculer le gradient de la fonction à optimiser. Dans la deuxième étape, on remplace la matrice hessienne B par une approximation (matrice H' [8x8]). Cette matrice est calculée en gardant une trace de la différence de gradients entre deux itérations consécutives $H_0=I$, soit la différence de gradient entre 4 itérations consécutives $H_0 = \nabla^2 f(x_0)$.
- **Choix du point de départ :** dans beaucoup de problèmes d'optimisation, le choix du point de départ est crucial pour que l'algorithme d'optimisation converge rapidement vers un minimum global. Le calcul du point de départ pour les problèmes de recherche linéaire "line search" s'effectue avec des algorithmes plus complexes que les algorithmes d'optimisation eux-mêmes. Dans le cas d'optimisation d'un convertisseur électrothermique (problème d'optimisation non linéaire à variables continues avec contraintes) le choix du point de départ n'a pas beaucoup d'influence sur la convergence de l'algorithme d'optimisation tant que les valeurs initiales du point de départ sont prises dans les domaines de définition. Néanmoins, nous avons validé chaque algorithme d'optimisation utilisé sur deux points différents de départ et les résultats de l'optimisation, vitesse de convergence et valeurs finales de l'optimisation étaient quasi identiques[9] [10] [11].

L'organigramme suivant de la Figure 6.13 illustre la méthodologie utilisée pour l'optimisation du convertisseur électrothermique :

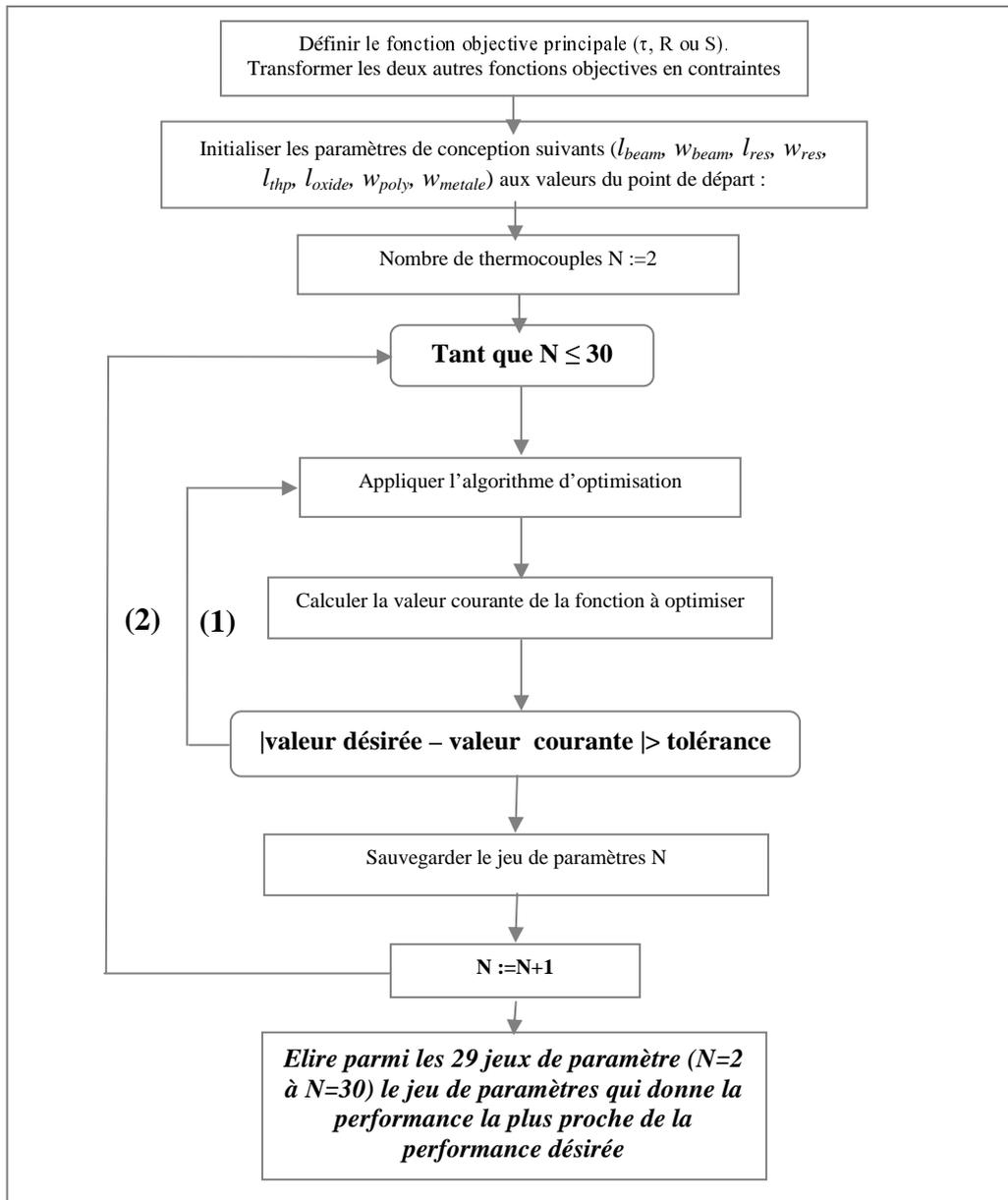


Figure 6.13. Étapes de l'optimisation d'un convertisseur électrothermique.

La première étape dans l'organigramme de la Figure 6.13 est de définir une fonction objective principale et de transformer les deux autres fonctions objectives en contraintes. Dans la suite, paragraphe expérimentations et résultats, nous allons procéder à trois modes d'optimisation :

- En gardant τ comme fonction objective principale et en transformant S et R en contraintes
- En gardant R comme fonction objective principale et en transformant S et τ en contraintes
- En gardant S comme fonction objective principale et en transformant R et τ en contraintes

Il faut noter ici que le fait de choisir une seule fonction objective comme moniteur d'optimisation ne change pas la nature du problème d'optimisation. Le fait de choisir S par exemple comme fonction objective principale, τ et R comme contraintes reste toujours une technique pour avoir le meilleur S, R et τ .

Une première analyse de ces trois modes (les fonctions objectives et les contraintes correspondantes) nous conduit à estimer que dans le cas où la surface S est la fonction objective principale, le problème de convergence sera plus persistant que dans les deux autres

cas. Cela vient du fait que la contrainte de “*facteur de forme*” impose une condition linéaire sur la surface S ($S = l_{beam} w_{beam}$, “*facteur de forme*” = l_{beam} / w_{beam}).

A noter, que dans la boucle interne (boucle 1), on optimise une fonction objective pour un nombre de thermocouples donné ($1 < N < 30$), et la boucle externe (boucle 2) fait varier ce nombre. L'étape finale de cet algorithme consiste à choisir parmi les jeux de paramètres résultant de la boucle 2, celui qui donne la meilleure performance ; la meilleure performance étant définie par la distance minimale entre la performance désirée et la performance obtenue.

6.4.2.1 Premier mode : la constante de temps τ comme fonction objective

Nous avons commencé le processus d'optimisation par fixer les valeurs suivantes ($\tau = 35e-3$ secondes, $S = 5000 \mu m^2$, $R = 85e-2$ volts) comme valeurs désirées finales des fonctions objectives. Ensuite nous avons transformé les fonctions objectives R et S en contraintes en les pondérant par des minima et maxima proches des valeurs désirées. Le problème d'optimisation en utilisant ce mode est formulé de la façon suivante :

$$\left. \begin{array}{l} \text{maximiser } \tau \\ \text{sous les contraintes initiales :} \\ \text{et les contraintes supplémentaires :} \end{array} \right\} \begin{cases} 100e-6 \leq l_{beam} \leq 500e-6 \\ 30e-6 \leq w_{beam} \leq 300e-6 \\ 10e-6 \leq l_{res} \leq 300e-6 \\ 0.8e-6 \leq w_{res} \leq 10e-6 \\ 20e-6 \leq l_{thp} \leq 500e-6 \\ 8e-6 \leq l_{oxide} \leq 500e-6 \\ 0.9e-6 \leq w_{poly} \leq 10e-6 \\ 0.8e-6 \leq w_{metal} \leq 10e-6 \\ 1 \leq l_{beam} / w_{beam} \leq 5 \\ 2 \leq N \leq 30 \\ 80 \leq R \leq 90 \\ 4000 \leq S \leq 5000 \end{cases}$$

Notons que le terme « minimiser τ » est utilisé dans le sens de trouver la valeur la plus proche de la valeur fixée ($\tau = 35e-3$), c'est-à-dire minimiser la fonction $|\tau - 35|$.

Nous avons appliqué les deux algorithmes d'optimisation détaillés dans l'annexe D, nous avons commencé par l'algorithme du gradient de la plus forte pente descendante vu la simplicité relative pour l'implémenter, nous avons procédé ensuite par l'algorithme BFGS.

I. Première méthode : algorithme du gradient de la plus forte pente descendante

Pour démarrer l'optimisation avec cet algorithme, nous avons choisi comme valeurs initiales du point de départ les valeurs données par le Tableau 6.3 Les seules mesures que nous avons prises dans le choix de ces valeurs initiales étaient de respecter leurs domaines de définition.

Point de départ		
Var.	Description	Valeur
L_{beam}	Longueur de la poutre	120.0e-6
W_{beam}	Largeur de la poutre	35.0e-6
L_{res}	Longueur de la résistance	33.0e-6
W_{res}	Largeur de la résistance	3e-6
W_{poly}	Largeur de Poly	2.4e-6
W_{metal}	Largeur de Metal	1.4e-6
L_{thp}	Longueur du thermocouple	80e-6
W_{oxide}	Largeur de Oxide	17e-6
N	Nombre de thermocouple	10
L_{beam}/W_{beam}	Facteur de forme	3.4

Tableau 6.3. Paramètres de conception initialisés aux valeurs du point de départ pour l'algorithme de la plus forte pente descendante.

Pour illustrer la convergence (ou la divergence) de l'algorithme utilisé, nous avons tracé les points correspondants aux itérations nécessaires pour arriver à la performance désirée (τ en fonction de S comme abscisse et R comme ordonnée). Il faut noter que le graphe tracé correspond à la séquence des itérations choisie à la fin du processus d'optimisation et identifié par le nombre de thermocouples N.

La Figure 6.14 montre le graphe de convergence de gradient de la plus forte pente descendante. Le graphe correspond à la séquence d'itérations effectuées par l'algorithme pour un nombre de thermocouples N=12. Ce nombre a été choisi par le processus de l'optimisation dans la phase finale, voir l'organigramme de la Figure 6.13.

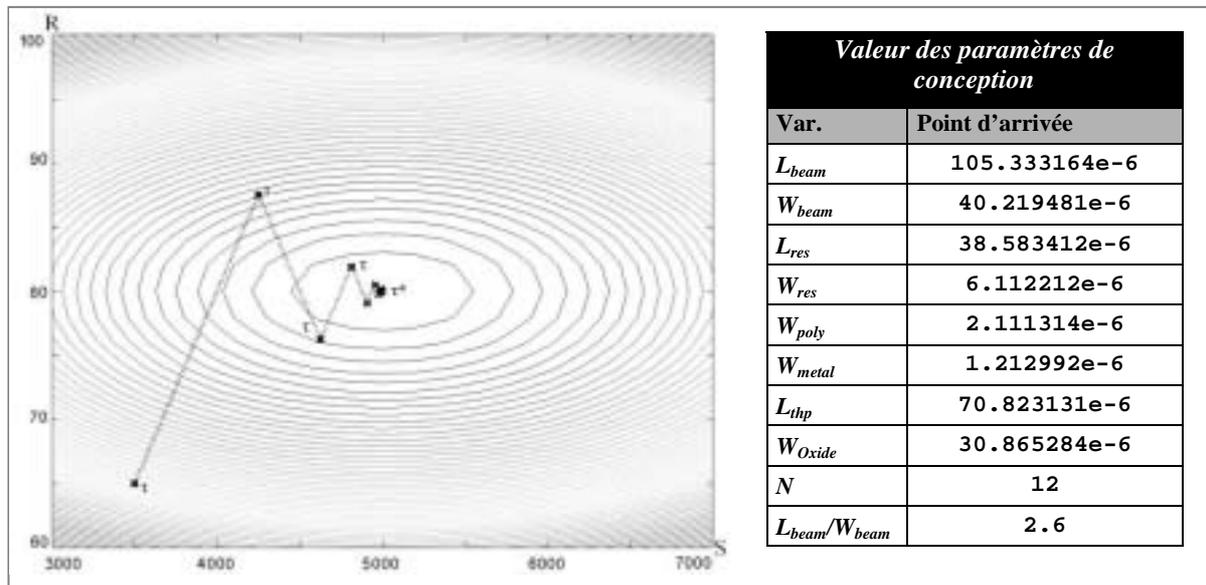


Figure 6.14. Graphe de convergence en utilisant l'algorithme de la plus forte pente descendante, et les valeurs finales de l'optimisation (τ comme fonction objective principale).

Notons que pour des valeurs de N inférieures à 8, l'algorithme n'arrive pas à converger. Deux raisons peuvent être envisagées pour expliquer ce phénomène :

- Physiquement, le fait d'avoir un nombre faible de thermocouples donne une réponse médiocre du composant, ce qui ramène le point de convergence en dehors du domaine de définition de R.
- Mathématiquement, l'algorithme n'est pas apte à converger vers des minima locaux. Cette hypothèse sera vérifiée en utilisant un autre algorithme d'optimisation.

Pour des valeurs de N comprises entre 8 et 30, l'algorithme converge vers une solution en moyenne de 5 à 7 itérations pour chaque boucle. Le graphe de la Figure 6.14 correspond à la séquence de 5 itérations générées par l'algorithme d'optimisation pour un nombre optimal de thermocouples N=12.

Nous avons généré le modèle comportemental correspondant aux paramètres optimisés de conception en utilisant la méthodologie de synthèse détaillée dans la première partie de ce chapitre. Les résultats de simulation de ce modèle sont montrés dans la Figure 6.15 :

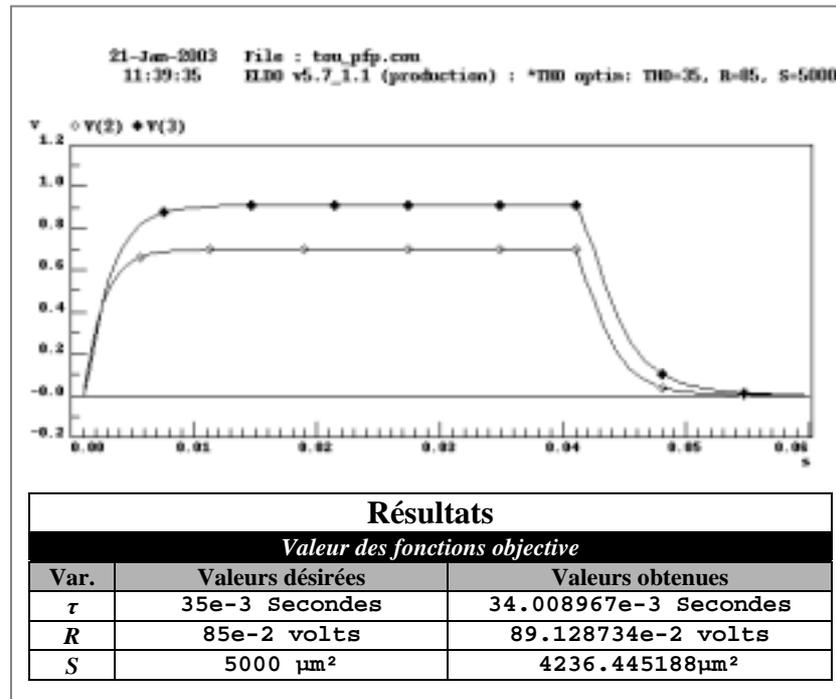


Figure 6.15. Résultats de simulation du comportement du convertisseur avant l'optimisation : (courbe v(2)) et après l'optimisation : (courbe v(3)) (τ comme fonction objective principale, l'algorithme de la plus forte pente descendant) et valeurs finales des fonctions objectives.

Les deux courbes v(2) et v(3) (graphe de la Figure 6.15) présentent la simulation de la réponse du convertisseur avant et après l'optimisation. Notons que l'on arrive à avoir plus de précision sur la valeur de τ que les autres fonctions. C'est dû au fait que τ est la fonction objective principale du problème.

II. Deuxième méthode : l'algorithme BFGS, $H_0=I$

Nous avons pris comme point de départ le même point utilisé dans l'algorithme précédent. Les valeurs initiales des différents paramètres de conception sont données par le Tableau 6.3. Comme valeurs finales des fonctions objectives, nous avons gardé les mêmes valeurs que l'algorithme précédent ($\tau=35\text{e-}3$ secondes, $R=85\text{e-}2$ volts, $S=5000 \mu\text{m}^2$).

L'algorithme BFGS a été utilisé comme méthode d'optimisation avec une approximation de premier ordre (Matrice hessienne calculée en gardant une trace de la différence de gradients entre deux itérations consécutives $H_0=I$).

La Figure 6.16 montre la convergence de l'algorithme BFGS vers un point de performance optimale ($\tau^*=35.001800\text{e-}3$ secondes, $S^*=4998.351754 \mu\text{m}^2$, $R^*=82.398142\text{e-}2$ volts). Le graphe correspond à une projection dans le plan (S, R) de la séquence d'itérations effectuées par l'algorithme pour un nombre de thermocouples N=15. Ce nombre a été choisi par le processus de l'optimisation dans la phase finale, voir l'organigramme de la Figure 6.13.

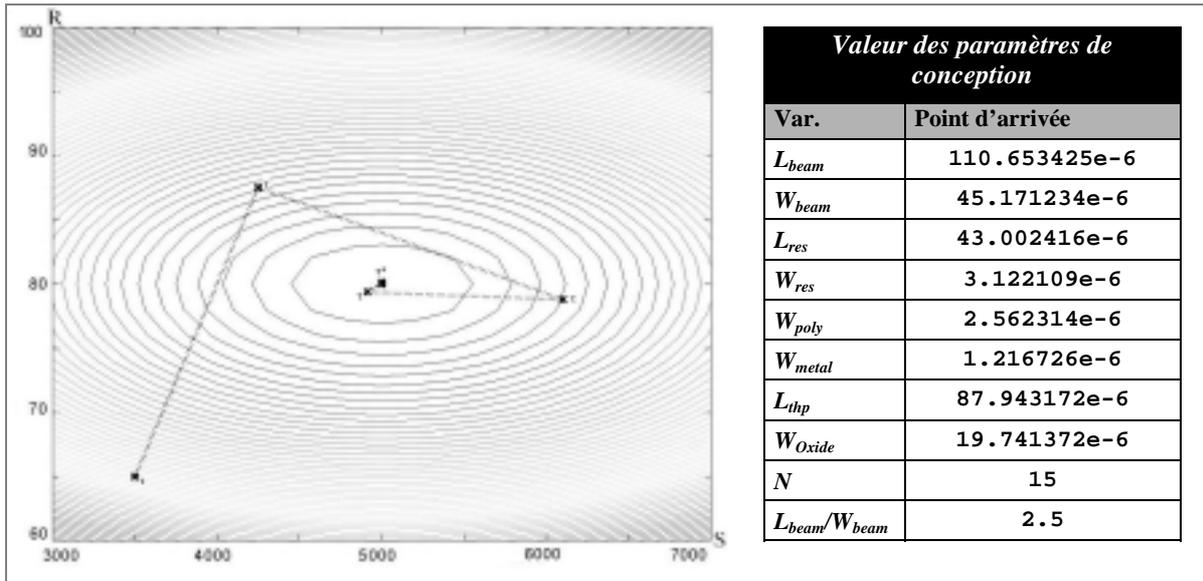


Figure 6.16. Graphe de convergence en utilisant la méthode BFGS, $H_0=I$, et valeurs finales de l'optimisation (τ comme fonction objective principale).

Cet algorithme converge vers une solution à partir de nombre de thermocouples $N=6$ (8 dans l'algorithme précédent), mais avec un coût très élevé (800 itérations pour $N=6$, 600 itérations pour $N=7$), à partir de $N=10$ l'algorithme commence à être économique au niveau du temps de calcul. Le graphe dans la Figure 6.16 correspond à la séquence de 4 itérations générées par l'algorithme d'optimisation pour un nombre optimal de thermocouples $N=15$.

Un modèle comportemental a été généré à partir des paramètres optimisés dont les valeurs sont données dans le tableau de la Figure 6.16 Les résultats de la simulation de ce modèle sont montrés dans la Figure 6.17 :

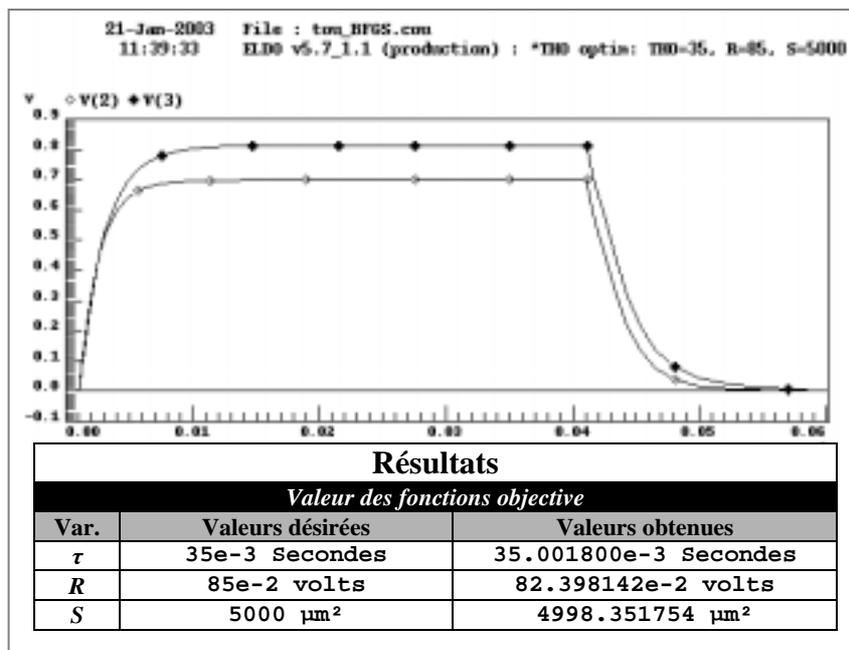


Figure 6.17. Résultats de simulation du comportement du convertisseur avant l'optimisation : (courbe $v(2)$) et après l'optimisation : (courbe $v(3)$) (τ comme fonction objective principale, l'algorithme BFGS $H_0=I$ comme méthode d'optimisation), et valeurs finales des fonctions objectives.

Les valeurs des fonctions objectives obtenues par cet algorithme sont très proches des valeurs désirées, voir le tableau de la Figure 6.17. Une comparaison entre les deux algorithmes utilisés sera discutée dans le paragraphe suivant.

III. Comparaison des deux méthodes

Au niveau de précision, les deux algorithmes donnent des solutions satisfaisantes (voir le tableau comparatif de la Figure 6.18), avec une légère supériorité pour l'algorithme BFGS.

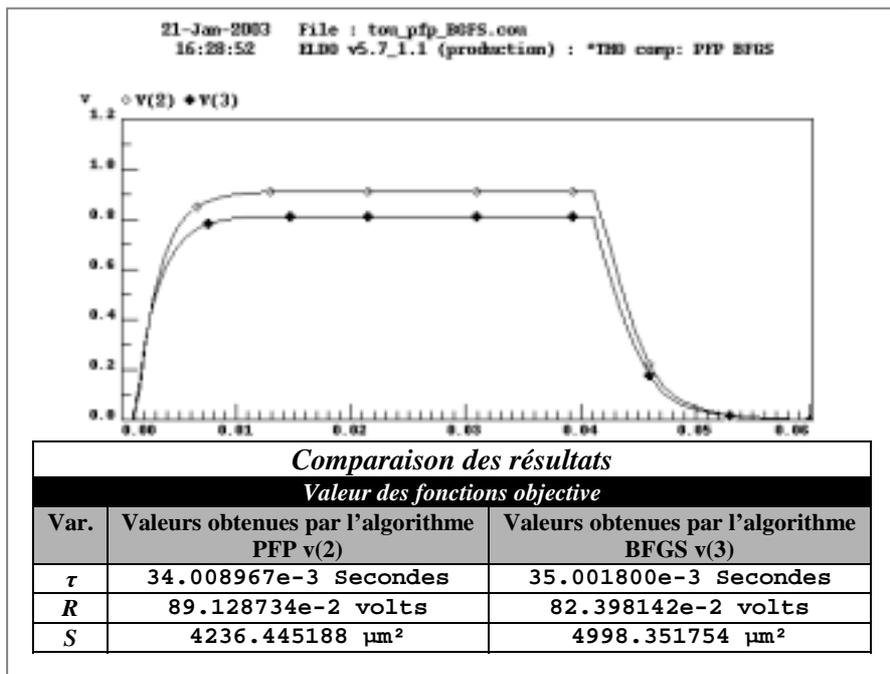


Figure 6.18. Tableau comparatif des résultats de simulation obtenus par les deux méthodes et les valeurs correspondantes des trois fonctions objectives.

L'algorithme BFGS nous a permis d'élargir le domaine de convergence ; convergence à partir de $N=6$ contre une convergence à partir de $N=8$ pour l'algorithme du gradient, mais à coût très élevé et pour trouver une solution qui ne sera pas retenue à la fin du cycle sur N . A la fin de bouclage, c'est la solution correspondante à $N=15$ qui a été retenue.

En terme de vitesse de convergence, l'algorithme BFGS est plus rapide que l'algorithme du gradient à condition de se trouver dans des zones proches du N optimal.

Nous pensons que dans le cas où il n'y a pas d'itérations sur N , nombre de thermocouples fixe, l'algorithme BFGS est le mieux adapté pour ce mode d'optimisation.

6.4.2.2 Deuxième mode : la réponse R comme fonction objective

Le deuxième mode que l'on a choisi comme processus d'optimisation a été de garder R comme fonction objective principale et de transformer les fonctions objectives τ et S en contraintes en les pondérant par des minima et maxima proches des valeur désirées. Nous avons fixé comme valeurs désirées finales des fonctions objectives, les valeurs suivantes ($R=90\text{e-}2$ volts, $S= 5500 \mu\text{m}^2$, $\tau=45\text{e-}3$ secondes). Le problème d'optimisation en utilisant ce mode est formulé de la façon suivante :

$$\begin{array}{l}
 \text{Maximiser R} \left\{ \begin{array}{l}
 \text{sous les contraintes initiales :} \\
 \left. \begin{array}{l}
 100e-6 \leq l_{beam} \leq 500e-6 \\
 30e-6 \leq w_{beam} \leq 300e-6 \\
 10e-6 \leq l_{res} \leq 300e-6 \\
 0.8e-6 \leq w_{res} \leq 10e-6 \\
 20e-6 \leq l_{thp} \leq 500e-6 \\
 8e-6 \leq l_{oxide} \leq 500e-6 \\
 0.9e-6 \leq w_{poly} \leq 10e-6 \\
 0.8e-6 \leq w_{metal} \leq 10e-6 \\
 1 \leq l_{beam} / w_{beam} \leq 5 \\
 2 \leq N \leq 30
 \end{array} \right\} \\
 \text{et les contraintes supplémentaires :} \\
 \left. \begin{array}{l}
 40 \leq \tau \leq 50 \\
 5300 \leq S \leq 6300
 \end{array} \right\}
 \end{array}
 \right.
 \end{array}$$

A noter que le terme « maximiser R » revient toujours à minimiser la fonction |R-90|.

Nous avons appliqué trois algorithmes pour résoudre ce mode d'optimisation. Nous avons commencé par l'algorithme du gradient de la plus forte pente descendante, ensuite nous avons procédé à l'algorithme BFGS avec son variant $H_0=I$ pour finir par le troisième algorithme qui est l'algorithme BFGS avec son variant $H_0 = \nabla^2 f(x_0)$.

I. Première méthode : algorithme du gradient de la plus forte pente descendante

Pour démarrer ce mode d'optimisation avec cet algorithme, nous avons choisi comme valeurs initiales du point de départ les valeurs données par le Tableau 6.4, nous avons choisi ces valeurs de manière à respecter les domaines de définition de ces valeurs initiales :

Point de départ		
Var.	Description	Valeur
L_{beam}	Longueur de la poutre	105.0e-6
W_{beam}	Largeur de la poutre	30.0e-6
L_{res}	Longueur de la résistance	28.0e-6
W_{res}	Largeur de la résistance	2e-6
W_{poly}	Largeur de Poly	2.4e-6
W_{metal}	Largeur de Metal	1.4e-6
L_{thp}	Longueur du thermocouple	85e-6
W_{oxide}	Largeur de Oxide	18e-6
N	Nombre de thermocouple	8
L_{beam}/W_{beam}	Facteur de forme	3.5

Tableau 6.4. Paramètres de conception initialisés aux valeurs du point de départ pour l'algorithme de la plus forte pente descendante.

La Figure 6.19 montre la convergence de l'algorithme du gradient de la plus forte pente descendante vers un point de performance optimale ($R^*=85.332145e-2$ volts, $S^*=6007.595697 \mu m^2$, $\tau^*=42.487524e-3$ Secondes). Le graphe correspond à une projection dans le plan (τ, R) de la séquence d'itérations effectuées par l'algorithme pour un nombre

de thermocouples $N=11$. Ce nombre a été choisi par le processus de l'optimisation dans la phase finale, voir l'organigramme de la Figure 6.13.

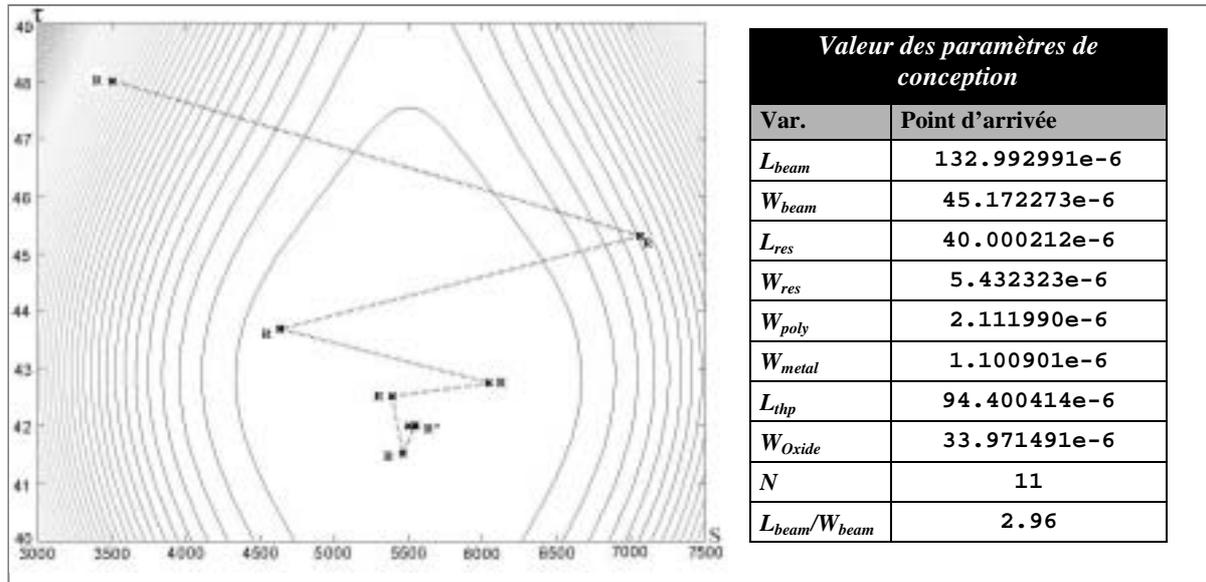


Figure 6.19. Graphe de convergence en utilisant l'algorithme de la plus forte pente descendante, et valeurs finales de l'optimisation (R comme fonction objective principale).

Dans ce mode d'optimisation, cet algorithme converge vers une solution pour un nombre de thermocouples N compris entre 8 et 15, un nombre faible de thermocouples rend la réponse du convertisseur médiocre, par contre un nombre élevé de thermocouples diminue la constante de temps.

En terme de nombre d'itérations, le graphe de la Figure 6.19 correspond à la séquence de 26 itérations générées par l'algorithme d'optimisation pour un nombre optimal de thermocouples $N=11$. Il faut noter que le temps de calcul en ce mode et avec cet algorithme devient très lourd (≈ 2 secondes par itération), c'est dû d'un côté à la difficulté de trouver un minimum local dans les domaines de définition, et d'un autre côté, au coût de calcul de la matrice hessienne qui devient important.

Notons aussi que contrairement à ce que l'on estimait, la séquence correspondante à $N=11$ est la séquence d'itérations la plus courte (26 itérations), le nombre d'itérations des autres séquences varie entre 45 et 105 itérations.

Un modèle comportemental a été généré à partir des paramètres optimisés dont les valeurs sont données dans le tableau de la Figure 6.19. Les résultats de la simulation de ce modèle sont montrés dans la Figure 6.20 :

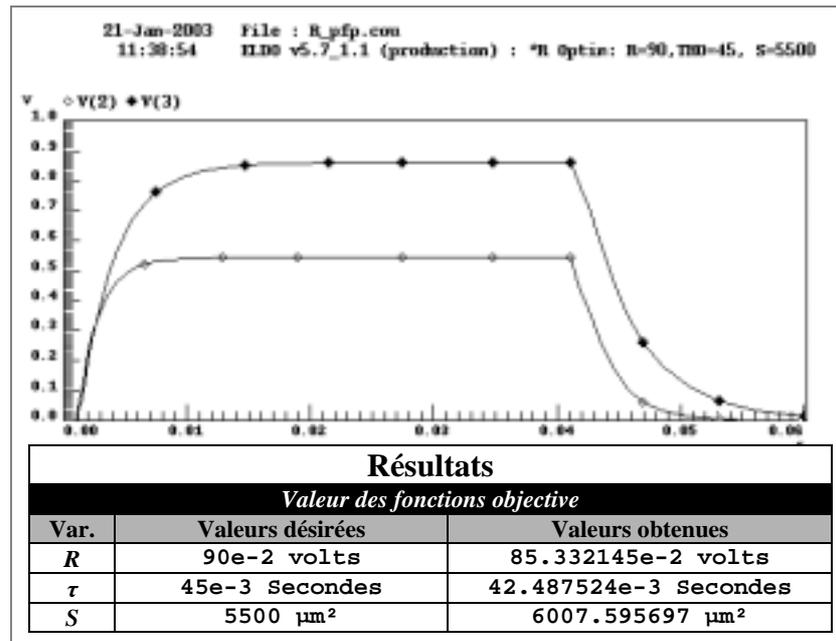


Figure 6.20. Résultats de simulation du comportement du convertisseur avant l'optimisation: (courbe $v(2)$) et après l'optimisation: (courbe $v(3)$) (R comme fonction objective principale, l'algorithme de la plus forte pente descendant), les valeurs finales des fonctions objectives.

Notons que la précision de la performance obtenue par rapport à la performance désirée, voir le tableau de la Figure 6.20 n'est pas très bonne, de plus si on ajoute le temps de calcul élevé pour converger vers la performance désirée, le recours aux autres algorithmes nous semble inévitable.

II. Deuxième méthode : l'algorithme BFGS, $H_0=I$

Nous avons pris comme point de départ le même point utilisé dans l'algorithme précédent. Les valeurs initiales des différents paramètres de conception sont données par le Tableau 6.4. Comme valeurs finales des fonctions objectives, nous avons gardé les mêmes valeurs que l'algorithme précédent ($R=90\text{e-}2$ volts, $S=5500 \mu\text{m}^2$, $T=45\text{e-}3$ secondes).

L'algorithme BFGS a été utilisé comme méthode d'optimisation avec une approximation de premier ordre (Matrice hessienne calculée en gardant une trace de la différence de gradients entre deux itérations consécutives $H_0=I$), le premier avantage que nous avons espéré était la réduction du temps de calcul en remplaçant la matrice hessienne par une approximation.

La Figure 6.21 montre la convergence de l'algorithme BFGS $H_0=I$ vers un point de performance optimale ($R^*=92.001221\text{e-}2$ volts, $S^*=5503.930732 \mu\text{m}^2$, $T^*=42.500501\text{e-}3$ secondes). Le graphe tracé correspond à une projection dans le plan (S,τ) de la séquence d'itérations effectuées par l'algorithme pour un nombre de thermocouples $N=14$. Ce nombre a été choisi par le processus de l'optimisation dans la phase finale, voir l'organigramme de la Figure 6.13.

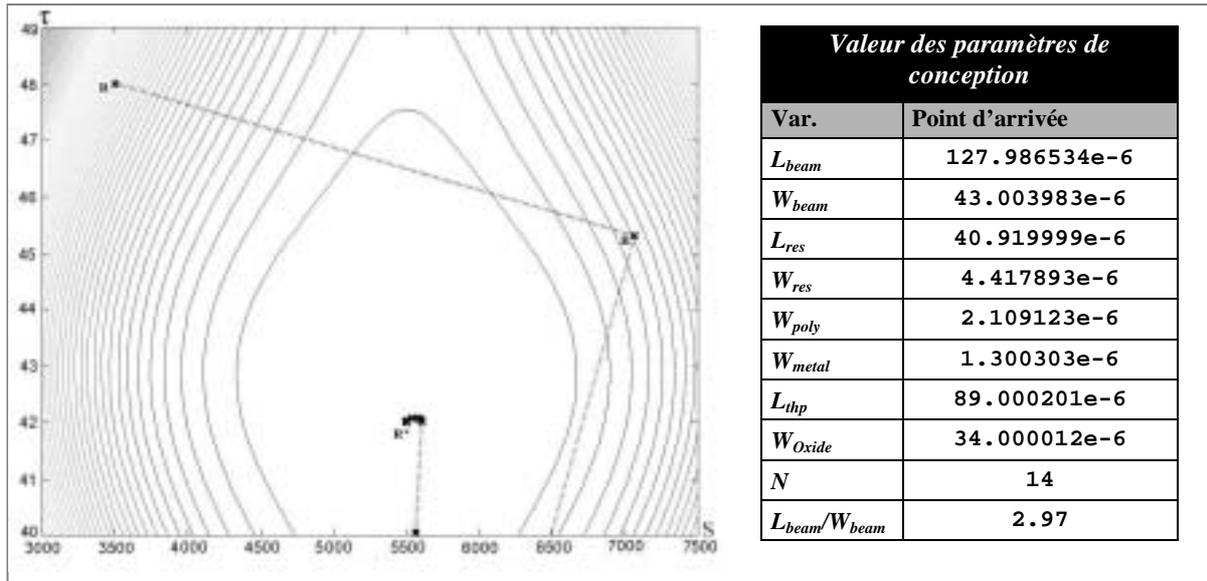


Figure 6.21. Graphe de convergence en utilisant la méthode BFGS, $H_0=I$, valeurs finales de l'optimisation (R comme fonction objective principale).

Cet algorithme converge vers une solution pour un nombre de thermocouples N compris entre 10 et 25 (8 à 15 dans l'algorithme précédent), le temps de calcul de chaque itération est moins important par rapport à l'algorithme précédent ($\approx 50e-3$ seconde par itération), mais avec un nombre d'itérations très élevé (≈ 2000 itérations pour $N=10$, ≈ 1000 itérations pour $N=25$). Le graphe de la Figure 6.21 correspond à la séquence de 1420 itérations générées par l'algorithme d'optimisation pour un nombre optimal de thermocouples $N=14$.

Les résultats de simulation du modèle généré en utilisant la méthodologie détaillée dans la première partie de ce chapitre sont illustrés dans la Figure 6.22.

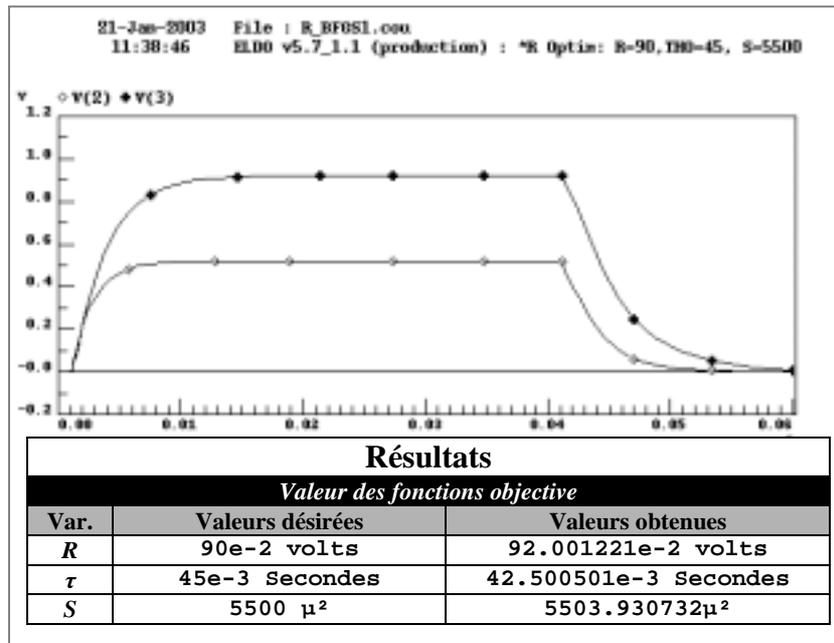


Figure 6.22. Résultats de simulation du comportement du convertisseur avant l'optimisation: (courbe $v(2)$) et après l'optimisation: (courbe $v(3)$) (R comme fonction objective principale, l'algorithme BFGS $H_0=I$), et valeurs finales des fonctions objectives.

La précision de la performance obtenue par cet algorithme par rapport à la performance désirée, voir le tableau comparatif de la Figure 6.22 est décevante. Ajoutons aussi le nombre d'itération nécessaire pour la convergence. Le seul avantage que l'on a obtenu en utilisant cette méthode était d'élargir le domaine de définition de N par rapport à l'algorithme précédent.

III. Troisième méthode : l'algorithme BFGS, $H_0 = \nabla^2 f(x_0)$

Nous avons pris comme point de départ le même point utilisé dans l'algorithme précédent. Les valeurs initiales des différents paramètres de conception sont données par le Tableau 6.4. Comme valeurs finales des fonctions objectives, nous avons gardé les mêmes valeurs que l'algorithme précédant ($R=90e-2$ volts, $S=5500 \mu m^2$, $\tau=45e-3$ secondes).

L'algorithme BFGS a été utilisé comme méthode d'optimisation avec une approximation de second ordre (Matrice hessienne calculée en gardant une trace de la différence de gradients entre quatre itérations consécutives $H_0 = \nabla^2 f(x_0)$).

La Figure 6.23 montre la convergence de l'algorithme BFGS $H_0 = \nabla^2 f(x_0)$ vers un point de performance optimale ($R^*=90.006677e-2$ volts, $S^*=5594.833135 \mu m^2$, $\tau^*=43.532187e-3$ secondes). Le graphe tracé correspond à une projection dans le plan (S, τ) de la séquence d'itérations effectuées par l'algorithme pour un nombre de thermocouples $N=12$. Ce nombre a été choisi par le processus de l'optimisation dans la phase finale, voir l'organigramme dans la Figure 6.13.

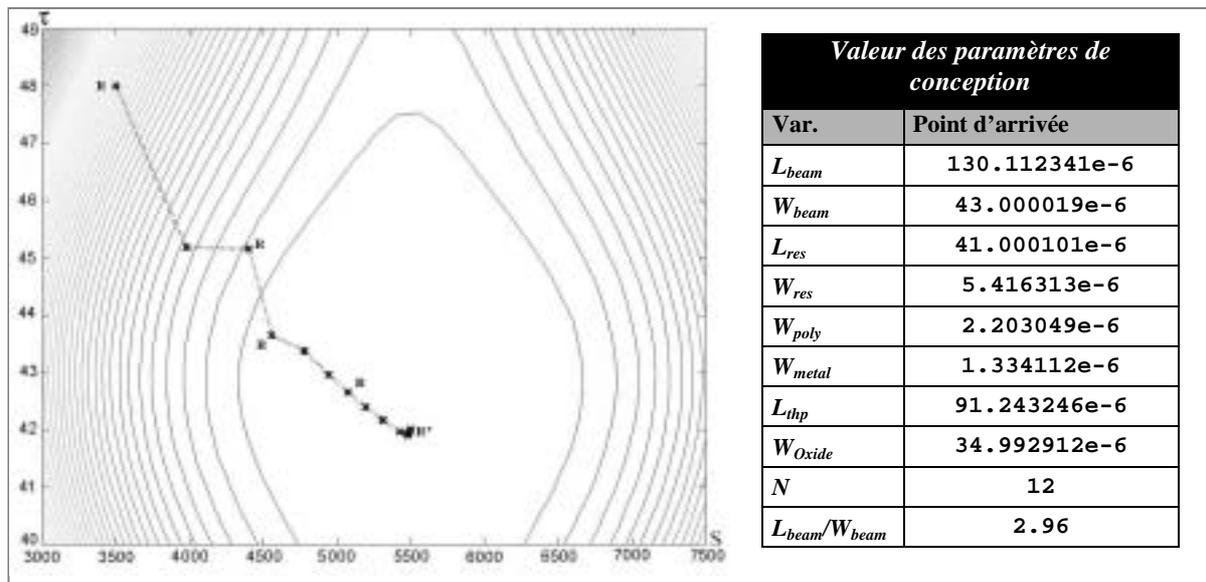


Figure 6.23. Graphe de convergence utilisant la méthode BFGS, $H_0 = \nabla^2 f(x_0)$ (graphe à gauche), valeurs finales de l'optimisation, R comme fonction objective principale.

L'algorithme converge vers un point de performance optimale pour toute valeur de N comprise entre 8 et 30. Le temps du calcul de chaque itération est plus important par rapport à l'algorithme précédent ($\approx 100e-3$ seconde par itération contre $\approx 50e-3$ pour l'algorithme BFGS $H_0=I$), mais il est beaucoup moins important que dans l'algorithme du gradient de la plus forte pente descendante (≈ 2 secondes par itération pour cet algorithme) et avec un nombre d'itérations très raisonnable (≈ 20 pour chaque boucle). Le graphe de la Figure 6.23 correspond à la séquence de 22 itérations générées par l'algorithme d'optimisation pour un nombre optimal de thermocouples $N=12$.

Avec le jeu de paramètres dont les valeurs sont données par le tableau de la Figure 6.23 nous avons généré le modèle comportemental optimisé du convertisseur. Les résultats de la simulation du composant avec ce modèle sont illustrés dans la Figure 6.24.

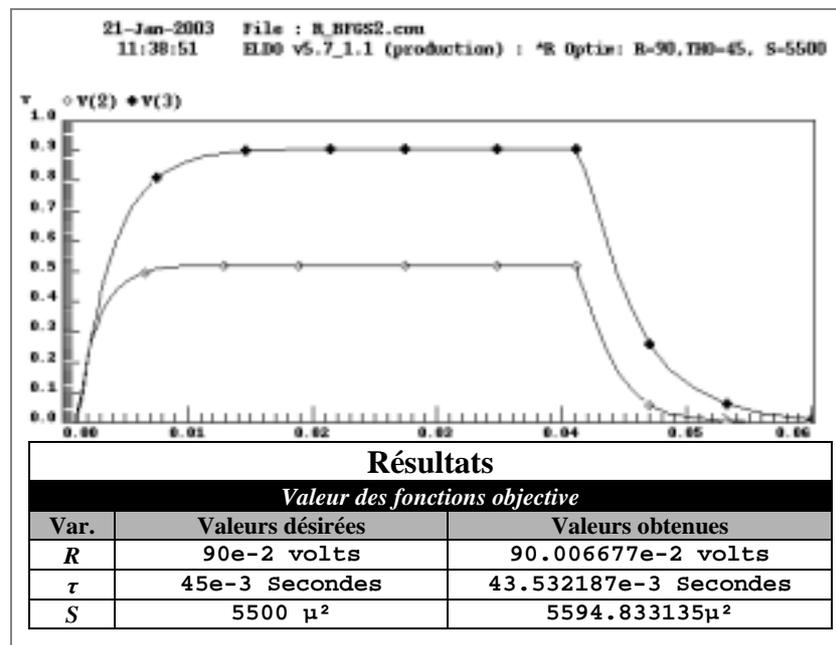


Figure 6.24. Résultats de simulation du comportement du convertisseur avant l'optimisation: (courbe $v(2)$) et après l'optimisation: (courbe $v(3)$) (R comme fonction objective principale, l'algorithme BFGS, $H_0 = \nabla^2 f(x_0)$), et valeurs finales des fonctions objectives.

Les résultats obtenus par cet algorithme sont très satisfaisants. En terme de précision, on arrive à avoir des valeurs très proches des valeurs de départ, voir le tableau comparatif de la Figure 6.24. Avec cet algorithme, la vitesse de la convergence s'est notamment améliorée, et on arrive à élargir de domaine de définition de N.

IV. Comparaison des trois méthodes

Avec ce mode d'optimisation (R comme fonction objective principale), le meilleur algorithme pour résoudre ce type de problème est sans doute l'algorithme BFGS avec son approximation de second ordre $H_0 = \nabla^2 f(x_0)$. Il donne une performance très proche de la performance désirée, voir le tableau comparatif des trois algorithmes de la Figure 6.25. En terme de temps de calcul il est un peu moins bon que l'algorithme BFGS $H_0=I$ ($\approx 100e-3$ seconde par itérations contre $\approx 50e-3$ seconde pour l'algorithme BFGS $H_0=I$), mais avec un nombre d'itération très raisonnable (≈ 20 itérations en moyenne par boucle contre ≈ 1500 itérations pour l'algorithme BFGS $H_0=I$ et ≈ 75 itérations pour l'algorithme du gradient de la plus forte pente descendante).

A noter que l'implémentation de cet algorithme n'est pas facile, l'approximation de second ordre est algorithmiquement difficile à réaliser.

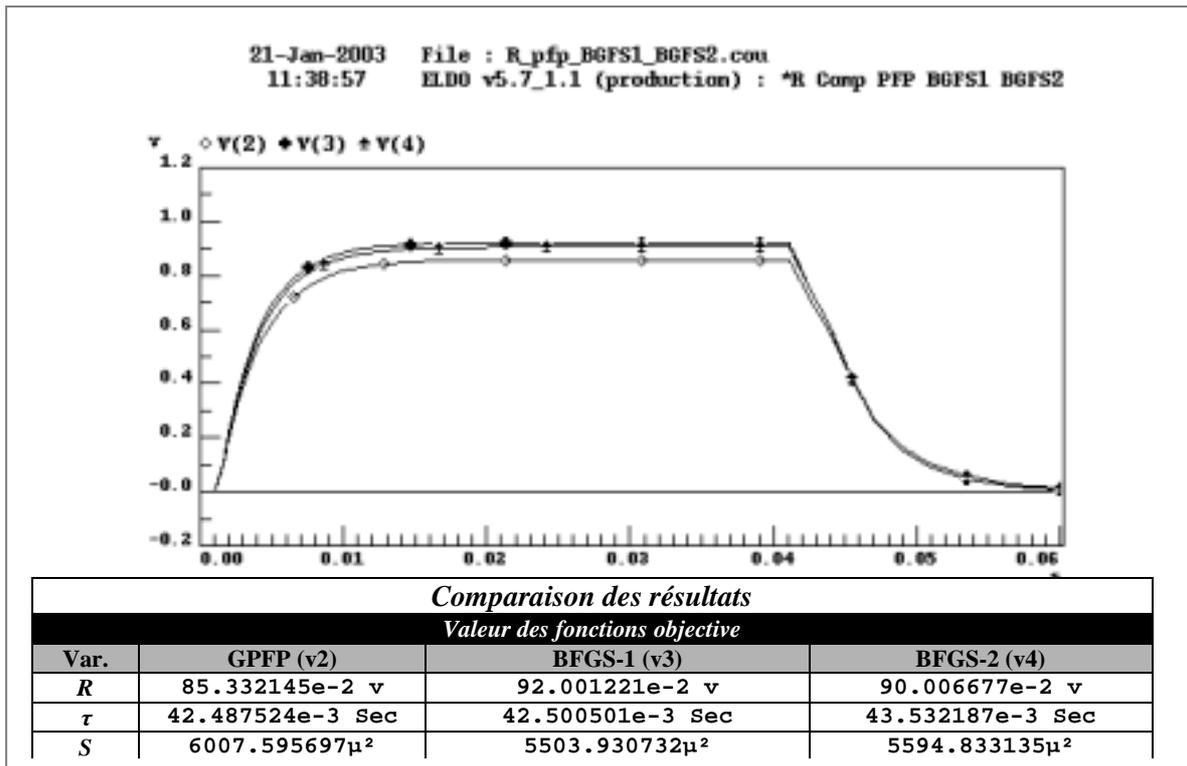


Figure 6.25. Tableau comparatif des résultats de simulation obtenus par les trois méthodes et les valeurs correspondantes des trois fonctions objectives.

6.4.2.3 Troisième mode : la surface S comme fonction objective

Le troisième mode que l'on a appliqué comme processus d'optimisation a été de garder S comme fonction objective principale et de transformer les fonctions objectives τ et R en contraintes en les pondérant par des minima et maxima proches des valeurs désirées. Nous avons fixé comme valeurs désirées finales des fonctions objectives, les valeurs suivantes ($R=75e-2$ volts, $S= 4000 \mu m^2$, $\tau=40e-3$ secondes). Le problème d'optimisation en utilisant ce mode est formulé de la façon suivante :

$$\begin{array}{l}
 \text{minimiser } S \left\{ \begin{array}{l}
 \text{sous les contraintes initiales :} \\
 \begin{array}{l}
 100e-6 \leq l_{beam} \leq 500e-6 \\
 30e-6 \leq w_{beam} \leq 300e-6 \\
 10e-6 \leq l_{res} \leq 300e-6 \\
 0.8e-6 \leq w_{res} \leq 10e-6 \\
 20e-6 \leq l_{thp} \leq 500e-6 \\
 8e-6 \leq l_{oxide} \leq 500e-6 \\
 0.9e-6 \leq w_{poly} \leq 10e-6 \\
 0.8e-6 \leq w_{metal} \leq 10e-6 \\
 1 \leq l_{beam} / w_{beam} \leq 5 \\
 2 \leq N \leq 30
 \end{array} \\
 \text{et les contraintes supplémentaires :} \\
 \begin{array}{l}
 35 \leq \tau \leq 45 \\
 70 \leq R \leq 80
 \end{array}
 \end{array}
 \right.
 \end{array}$$

A noter que le terme « minimiser S » revient à minimiser la fonction $|S-90|$.

Nous avons appliqué trois algorithmes pour résoudre ce mode d'optimisation. Nous avons commencé par l'algorithme du gradient de la plus forte pente descendante, ensuite nous avons procédé à l'algorithme BFGS avec son variant $H_0=I$ pour finir par le troisième algorithme qui est l'algorithme BFGS avec son variant $H_0 = \nabla^2 f(x_0)$.

I. Première méthode : algorithme du gradient de la plus forte pente descendante

Pour démarrer ce mode d'optimisation avec cet algorithme, nous avons choisi comme valeurs initiales de point de départ les valeurs données par le Tableau 6.5, nous avons choisi ces valeurs de manière à respecter les domaines de définition de ces valeurs initiales :

Point de départ		
Var.	Description	Valeur
L_{beam}	Longueur de la poutre	85.0e-6
W_{beam}	Largeur de la poutre	40.0e-6
L_{res}	Longueur de la résistance	30.0e-6
W_{res}	Largeur de la résistance	1.2e-6
W_{poly}	Largeur de Poly	2.4e-6
W_{metal}	Largeur de Metal	1.4e-6
L_{thp}	Longueur du thermocouple	83.8e-6
W_{oxide}	Largeur de Oxyde	10e-6
N	Nombre de thermocouple	6
L_{beam}/W_{beam}	Facteur de forme	2.13

Tableau 6.5. Paramètres de conception initialisés aux valeurs du premier point de départ pour l'algorithme de la plus forte pente descendante.

L'algorithme n'arrive pas à converger quelque soit la valeur de N , le nombre de thermocouples. Les graphes de la Figure 6.26 montre le phénomène de zigzag dont l'algorithme souffre pour ce mode d'optimisation :

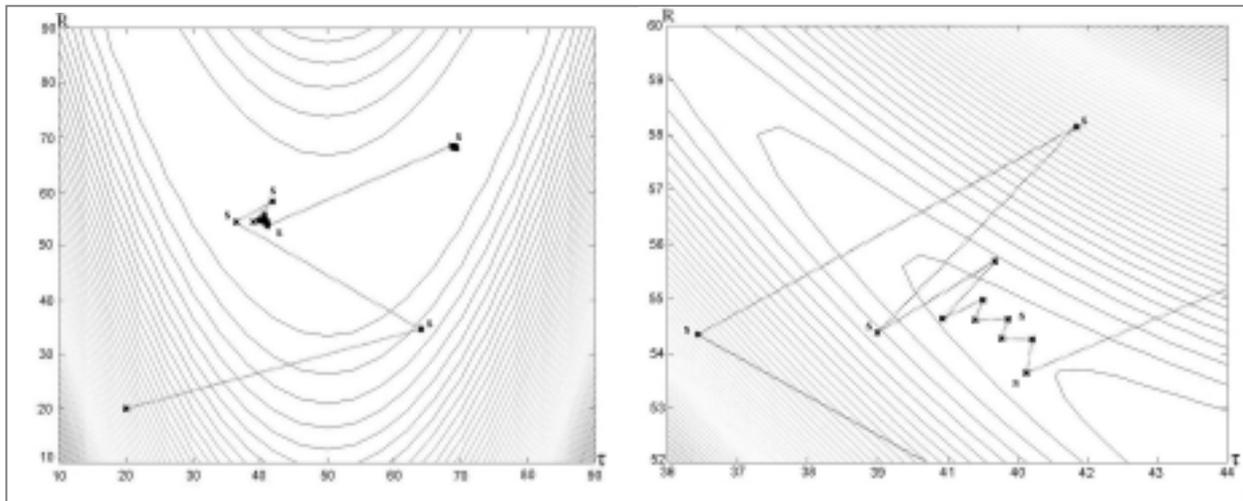


Figure 6.26. Résultats obtenus par la méthode de la plus forte pente descendante pour le premier point de départ où il n'y a pas de convergence (graphe en haut à gauche) et un zoom sur la zone de divergence montrant le phénomène de zigzag (en haut à droite).

Il y a deux hypothèses qui expliquent le phénomène de zigzag :

- Ce phénomène se produit quand le gradient de la fonction objective pour l'itération courante est très proche de l'inverse de celui de l'itération précédente. L'accumulation

des petites différences de gradient conduit à une périodicité dans le déplacement d'un point vers un autre. Une séquence de mêmes points se répète pendant la même boucle ce qui entraîne la divergence de l'algorithme.

- Ce phénomène est tout simplement le résultat d'un mauvais choix du point de départ.

Pour écarter la deuxième hypothèse, on a choisi un deuxième point de départ dont les valeurs des différents paramètres sont données par le Tableau 6.6 :

Point de départ		
Var.	Description	Valeur
L_{beam}	Longueur de la poutre	$120.0e-6$
W_{beam}	Largeur de la poutre	$40.0e-6$
L_{res}	Longueur de la résistance	$30.0e-6$
W_{res}	Largeur de la résistance	$1.5e-6$
W_{poly}	Largeur de Poly	$2.4e-6$
W_{metal}	Largeur de Metal	$1.4e-6$
L_{thp}	Longueur du thermocouple	$98e-6$
W_{oxide}	Largeur de Oxyde	$10e-6$
N	Nombre de thermocouple	18
$\frac{L_{beam}}{W_{beam}}$	Facteur de forme	3

Tableau 6.6. Paramètres de conception initialisés aux valeurs du deuxième point de départ pour l'algorithme de la plus forte pente descendante.

Les graphes de la Figure 6.27 montrent que le changement de point de départ n'améliore rien à la divergence de cet algorithme. Le phénomène de zigzag se reproduit avec une séquence différente de points.

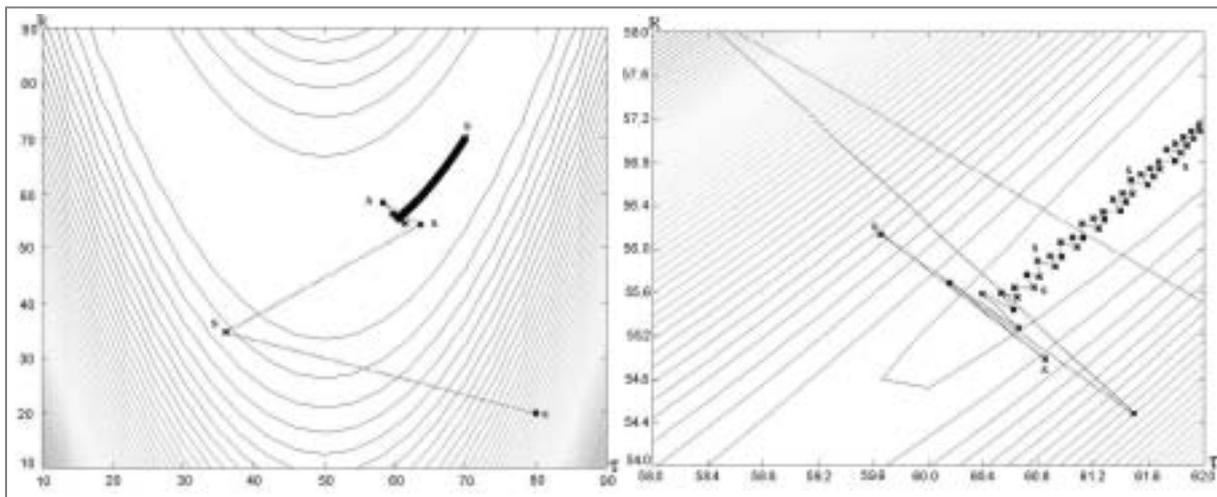


Figure 6.27. Résultats obtenus par la méthode de la plus forte pente descendante pour le deuxième point de départ où il n'y a pas de convergence (graphe en haut à gauche) et un zoom sur la zone de divergence montrant le phénomène de zigzag (en haut à droite).

Nous pensons que le problème de divergence de cet algorithme vient de la contrainte "facteur de forme" qui impose une condition linéaire sur S .

Pour obtenir une solution pour ce mode d'optimisation, on a recours aux algorithmes quasi newtoniens.

II. Deuxième méthode : l'algorithme BFGS, $H_0=I$

Nous avons pris comme point de départ le premier point utilisé dans l'algorithme précédent. Les valeurs initiales des différents paramètres de conception sont données par le Tableau 6.5. Comme valeurs finales des fonctions objectives, nous avons gardé les mêmes valeurs que l'algorithme précédent ($S= 4000 \mu\text{m}^2$, $R=75\text{e-}2 \text{ volts}$, $\tau=40\text{e-}3 \text{ secondes}$).

Une approximation de premier ordre (Matrice hessienne calculée en gardant une trace de la différence de gradients entre deux itérations consécutives $H_0=I$) de l'algorithme BFGS a été utilisée comme méthode d'optimisation.

La Figure 6.28 montre la convergence de l'algorithme BFGS $H_0=I$ vers un point de performance optimale ($s^*=4068.051571 \mu\text{m}^2$, $R^*=74.811231\text{e-}2 \text{ volts}$, $\tau^*=36.800352\text{e-}3 \text{ secondes}$). Le graphe tracé correspond à une projection dans le plan (R, τ) de la séquence d'itérations effectuées par l'algorithme pour un nombre de thermocouples $N=8$. Ce nombre a été choisi par le processus de l'optimisation dans la phase finale (voir l'organigramme de la Figure 6.13)

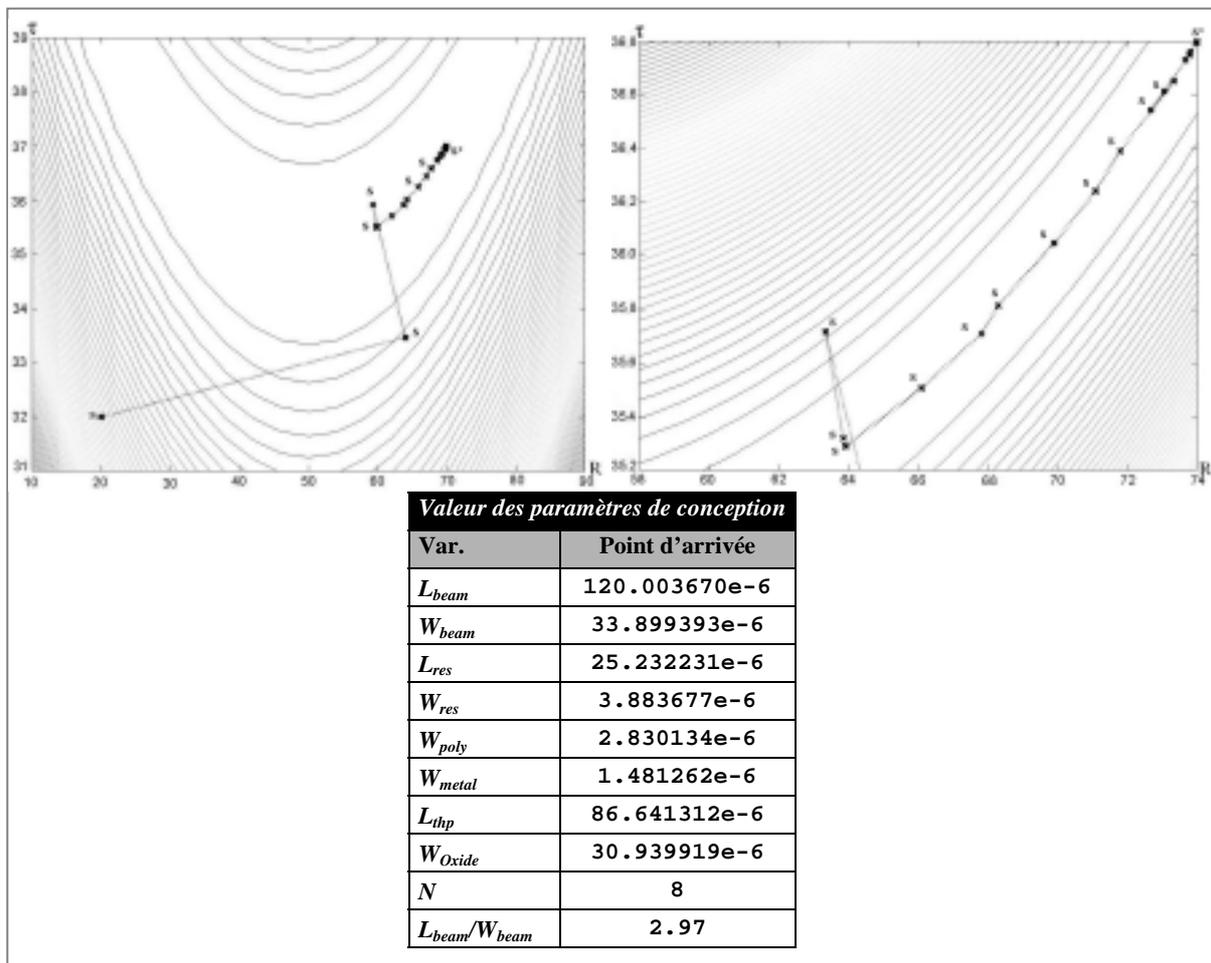


Figure 6.28. Graphe de convergence obtenu en utilisant la méthode BFGS, $H_0=I$ et en partant du premier point de départ, un zoom sur la zone de convergence, et valeurs finales de l'optimisation (S comme fonction objective principale).

Cet algorithme converge vers une solution pour un nombre de thermocouples $N = 8$ en 19 itérations avec un temps de calcul d'environ $200\text{e-}3$ seconde par itération. Pour les autres valeurs de N , on n'arrive pas à faire converger l'algorithme. Le graphe de la Figure 6.28

correspond à la seule séquence convergente de 19 itérations générées par l'algorithme d'optimisation pour un nombre optimal de thermocouples $N=8$.

Les résultats de simulation du modèle correspondant aux valeurs optimisées du tableau de la Figure 6.28 et généré en utilisant la méthodologie détaillée dans la première partie de ce chapitre sont illustrés dans la Figure 6.29.

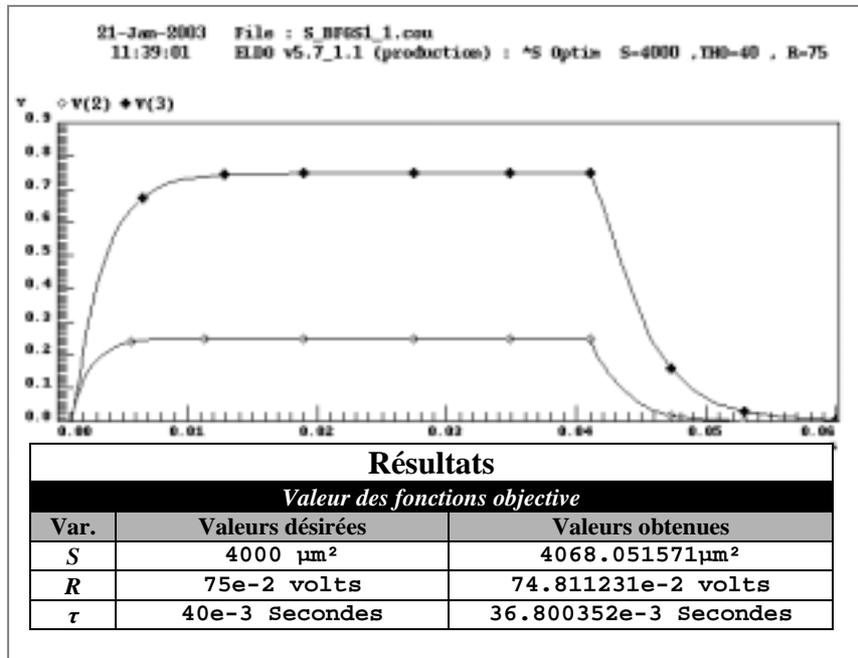


Figure 6.29. Résultats de simulation du comportement du convertisseur avant l'optimisation : (courbe $v(2)$) et après l'optimisation : (courbe $v(3)$) (S comme fonction objective principale, l'algorithme BFGS $H_0=I$, premier point de départ), et valeurs finales des fonctions objectives.

La précision de la performance obtenue par cet algorithme par rapport à la performance désirée (voir le tableau comparatif de la Figure 6.29) est satisfaisant en terme de minimisation de la surface et ($s^*=4068.051571 \mu\text{m}^2$ contre $s=4000 \mu\text{m}^2$.comme valeur désirée), nous sommes arrivé aussi à avoir une amplitude de réponse très proche de l'amplitude voulue.

Néanmoins, pour tester l'impact du choix de point de départ sur la convergence de l'algorithme d'une part, et sur la précision de la solution obtenue d'autre part, nous avons procédé à une autre série d'optimisation en gardant le même algorithme (BFGS $H_0=I$) comme méthode d'optimisation et les même valeurs ($S= 4000 \mu\text{m}^2$, $R=75\text{e-}2$ volts, $\tau=40\text{e-}3$ secondes) comme performance désirée et en changeant le point de départ vers le deuxième point de départ de l'algorithme précédent dont les valeurs des différents paramètres sont données par le Tableau 6.6.

La Figure 6.30 montre la convergence de l'algorithme BFGS $H_0=I$ avec ce deuxième point de départ vers un point de performance optimale ($s^*=4049.190008 \mu\text{m}^2$, $r^*=74.756413\text{e-}2$ volts, $\tau^*=37.001514\text{e-}3$ secondes). Le graphe tracé correspond à une projection dans le plan (R,τ) de la séquence d'itérations effectuées par l'algorithme pour un nombre de thermocouples $N=12$. Ce nombre a été choisi par le processus de l'optimisation dans la phase finale, voir l'organigramme de la Figure 6.13.

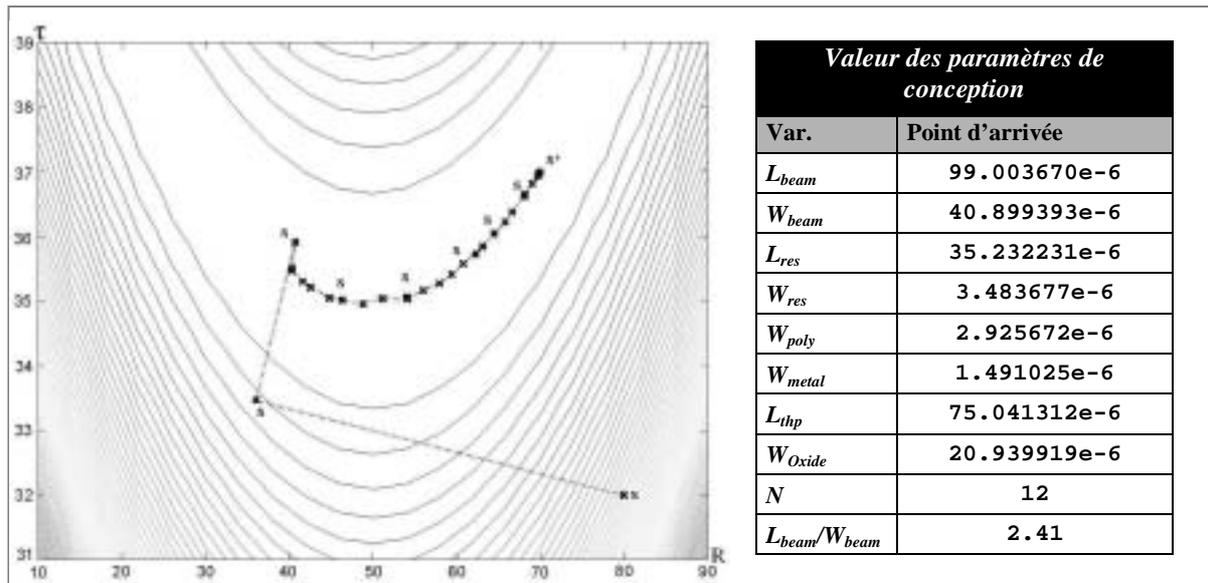


Figure 6.30. Graphe de convergence en utilisant la méthode BFGS, $H_0=I$ et en partant du deuxième point de départ, et valeurs finales de l'optimisation (S comme fonction objective principale).

Cet algorithme converge vers une solution pour deux nombres de thermocouples ($N=8$, $N=12$)

- Le premier N est celui obtenu dans la section précédente ($N=8$). Pour cette valeur de N l'algorithme converge en 27 itérations avec un temps de calcul d'environ $200e-3$ seconde par itération.
- Le deuxième N dont la séquence de convergence est tracé dans le graphe de la Figure 6.30 donne un nombre de thermocouple égal à 12. C'est la séquence qui donne la meilleure performance et qui a été retenue dans la phase finale de l'optimisation, voir l'organigramme de la Figure 6.13.

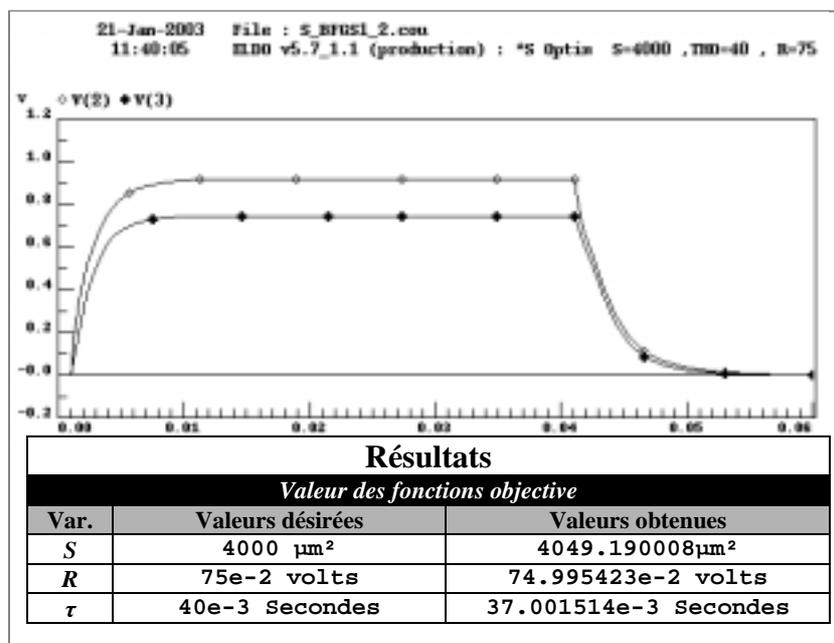


Figure 6.31. Résultats de simulation du comportement du convertisseur avant l'optimisation : (courbe $v(2)$) et après l'optimisation : (courbe $v(3)$) (S comme fonction objective principale, l'algorithme BFGS $H_0=I$, deuxième point de départ), et valeurs finales des fonctions objectives.

Pour $N=12$ l'algorithme converge vers une solution en 32 itérations avec un temps du calcul d'environ $200e-3$ seconde par itération. La précision de la performance obtenue par cet algorithme par rapport à la performance désirée, voir le tableau comparatif de la Figure 6.31 est satisfaisant en terme de minimisation de la surface et ($s^*=4049.190008 \mu\text{m}^2$ contre $s^*=4068.051571\mu\text{m}^2$.comme valeur obtenue en partant du premier point de départ), nous sommes arrivé aussi à avoir une amplitude de réponse très proche de l'amplitude voulue ($R^*=74.756413e-2 \text{ volts}$, contre $R^*= R^*=74.811231e-2 \text{ volts}$ en partant du premier point de départ) et une constante du temps légèrement améliorée ($\tau^*=37.001514e-3 \text{ Secondes}$ contre $\tau^*=36.800352e-3 \text{ Secondes}$ en partant de premier point).

Il faut noter que théoriquement, nous ne nous attendions pas à avoir un tel changement de géométrie du composant optimisé en changeant tout simplement le point de départ[9] [11]. Nous pensons qu'une approximation de premier ordre ne suffit pas pour résoudre le problème du gradient et du gradient inverse, cela nous a ramené à tester l'approximation de second ordre de l'algorithme BFGS.

III. Troisième méthode : l'algorithme BFGS, $H_0 = \nabla^2 f(x_0)$

Nous avons pris comme point de départ le premier point utilisé dans l'algorithme précédent. Les valeurs initiales des différents paramètres de conception sont données par le Tableau 6.5. Comme valeurs finales des fonctions objectives, nous avons gardé les mêmes valeurs que l'algorithme précédent ($S= 4000 \mu\text{m}^2$, $R=75e-2 \text{ volts}$, $\tau=40e-3 \text{ Secondes}$).

La Figure 6.32 montre la convergence de l'algorithme BFGS $H_0 = \nabla^2 f(x_0)$ avec ce premier point de départ vers un point de performance optimale ($s^*=4065.668255 \mu\text{m}^2$, $R^*=74.756413e-2 \text{ volts}$, $\tau^*=39.991212e-3 \text{ secondes}$). Le graphe tracé correspond à une projection dans le plan (R,τ) de la séquence d'itérations effectuées par l'algorithme pour un nombre de thermocouples $N=8$. Ce nombre a été choisi par le processus de l'optimisation dans la phase finale, voir l'organigramme de la Figure 6.13.

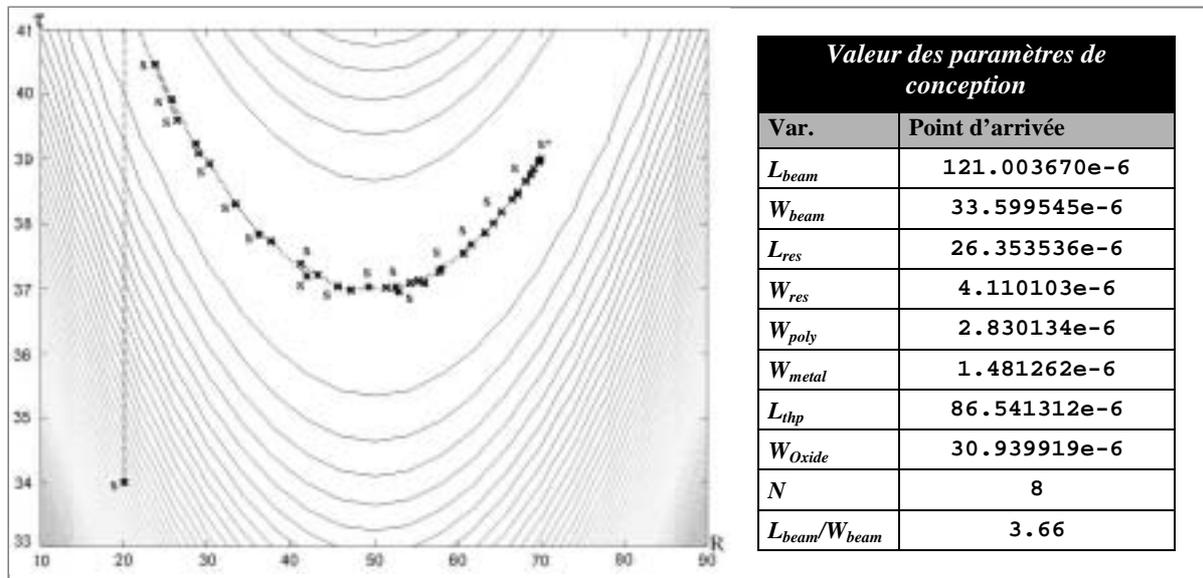


Figure 6.32. Graphe de convergence en utilisant la méthode BFGS, $H_0 = \nabla^2 f(x_0)$ et en partant du premier point de départ, et valeurs finales de l'optimisation (S comme fonction objective principale).

L'algorithme converge vers un point de performance optimale pour toute valeur de N comprise entre 8 et 20. Le temps du calcul de chaque itération est plus important par

rapport à l'algorithme précédent ($\approx 300e-3$ seconde par itération contre $\approx 50e-3$ pour l'algorithme BFGS $H_0=I$), avec un nombre d'itérations ≈ 50 pour chaque boucle. Le graphe de la Figure 6.32 correspond à la séquence de 52 itérations générées par l'algorithme d'optimisation pour un nombre optimal de thermocouples $N=8$.

Nous avons généré le modèle comportemental correspondant aux paramètres optimisés dont les valeurs sont données par le tableau de la Figure 6.32. Les résultats de simulation de ce modèle sont illustrés dans la Figure 6.33.

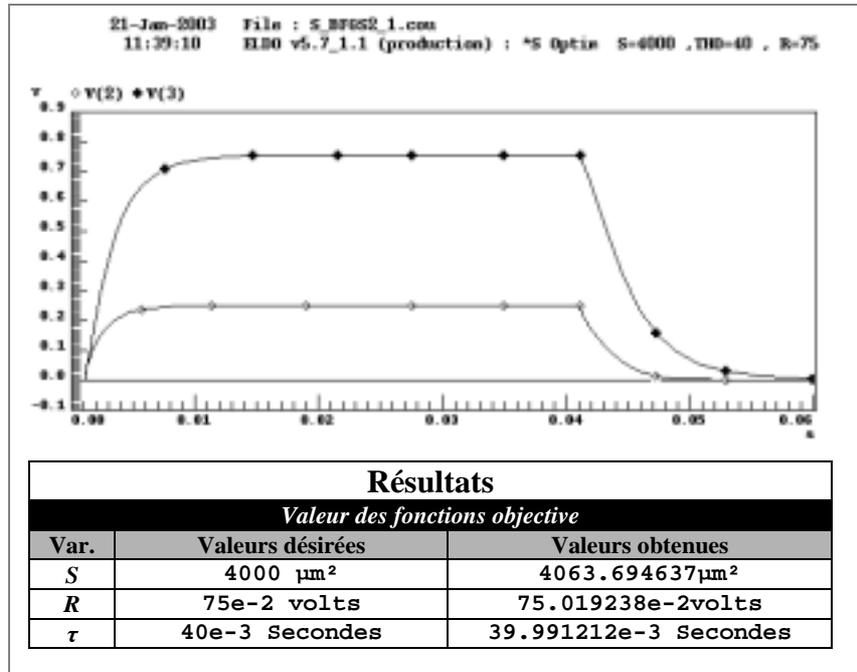


Figure 6.33. Résultats de simulation du comportement du convertisseur avant l'optimisation : (courbe $v(2)$) et après l'optimisation : (courbe $v(3)$) (S comme fonction objective principale, l'algorithme BFGS, $H_0 = \nabla^2 f(x_0)$, premier point de départ), et valeurs finales des fonctions objectives .

La précision de la performance obtenue par cet algorithme par rapport à la performance désirée , voir le tableau comparatif de la Figure 6.33, est optimale en terme de minimisation de la surface et ($s^*=4065.668255 \mu\text{m}^2$ contre $s=4000 \mu\text{m}^2$ comme valeur désirée), nous sommes arrivé aussi à avoir une amplitude de réponse très proche de l'amplitude voulue ($r^*=74.995423e-2$ volts contre $R=75e-2$ volts comme valeur désirée) et surtout une valeur de constante de temps quasi identique à la valeur désirée ($\tau^*=39.991212e-3$ Secondes contre $\tau=40e-3$ Secondes).

Pour se conformer aux estimations théoriques en vérifiant que le point de départ n'a pas d'effet ni sur la convergence de l'algorithme, ni sur la valeur de la performance optimale, nous avons procédé à une autre série d'optimisation en gardant le même algorithme (BFGS $H_0 = \nabla^2 f(x_0)$) comme méthode d'optimisation et les mêmes valeurs ($S= 4000 \mu\text{m}^2$, $R=75e-2$ volts, $\tau=40e-3$ secondes) comme performance désirée et en changeant le point du départ vers le deuxième point de départ de l'algorithme précédent dont les valeurs des différents paramètres sont données par le Tableau 6.6.

La Figure 6.34 montre la convergence de l'algorithme BFGS $H_0 = \nabla^2 f(x_0)$ avec ce deuxième point de départ vers un point de performance optimale ($s^*=4065.668255 \mu\text{m}^2$, $r^*=74.756413e-2$ volts, $\tau^*=39.825432e-3$ secondes). Le graphe tracé correspond à une

projection dans le plan (R, τ) de la séquence d'itérations effectuées par l'algorithme pour un nombre de thermocouples $N=8$. Ce nombre a été choisi par le processus de l'optimisation dans la phase finale, voir l'organigramme de la Figure 6.13.

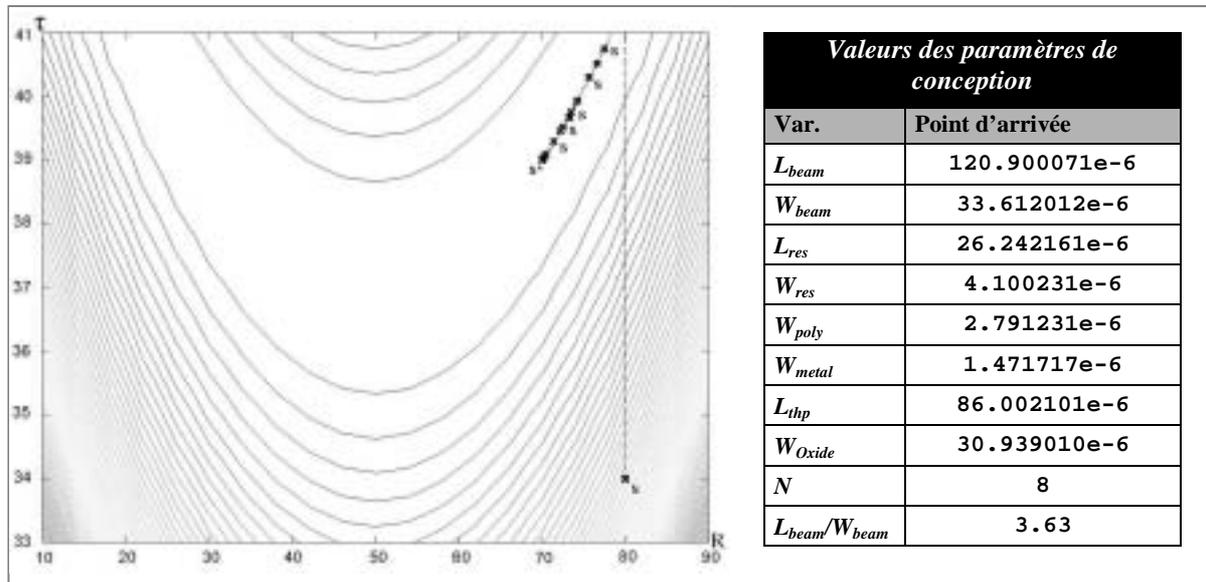


Figure 6.34. Graphe de convergence en utilisant la méthode BFGS, $H_0 = \nabla^2 f(x_0)$ et en partant du deuxième point de départ, et valeurs finales de l'optimisation (S comme fonction objective principale).

L'algorithme converge vers un point de performance optimale pour toute valeur de N compris entre 8 et 20. Le temps du calcul de chaque itération reste de l'ordre de $\approx 300e-3$ seconde par itération, avec un nombre d'itérations ≈ 50 pour chaque boucle. Le graphe de la Figure 6.34 correspond à la séquence de 48 itérations générées par l'algorithme d'optimisation pour un nombre optimal de thermocouples $N=8$.

En analysant les résultats d'optimisation obtenus en utilisant l'algorithme BFGS $H_0 = \nabla^2 f(x_0)$ pour les deux points de départ, on peut constater que le choix du point de départ n'a pas d'impact sur la convergence (divergence) de l'algorithme :

1. Le point de départ n'a pas d'effet sur le domaine de définition de N (les valeurs de N restent entre 8 et 20 pour les deux points de départ).
2. Le point de départ n'a pas d'effet sur le temps de calcul d'itération (toujours de l'ordre de $\approx 300e-3$ seconde par itération).
3. Le point de départ n'a pas d'effet sur le nombre d'itérations par boucle (le nombre d'itérations reste de l'ordre de 50 par boucle)

Notons aussi que les deux géométries obtenues en partant des deux points différents de départ sont quasi identiques (voir les tableaux de paramètres optimisés de la Figure 6.32 et la Figure 6.34).

Afin d'analyser la performance du composant optimisé, nous avons généré le modèle comportemental correspondant aux paramètres optimisés dont les valeurs sont données par le tableau de la Figure 6.34. Les résultats de simulation de ce modèle sont illustrés dans la Figure 6.35.

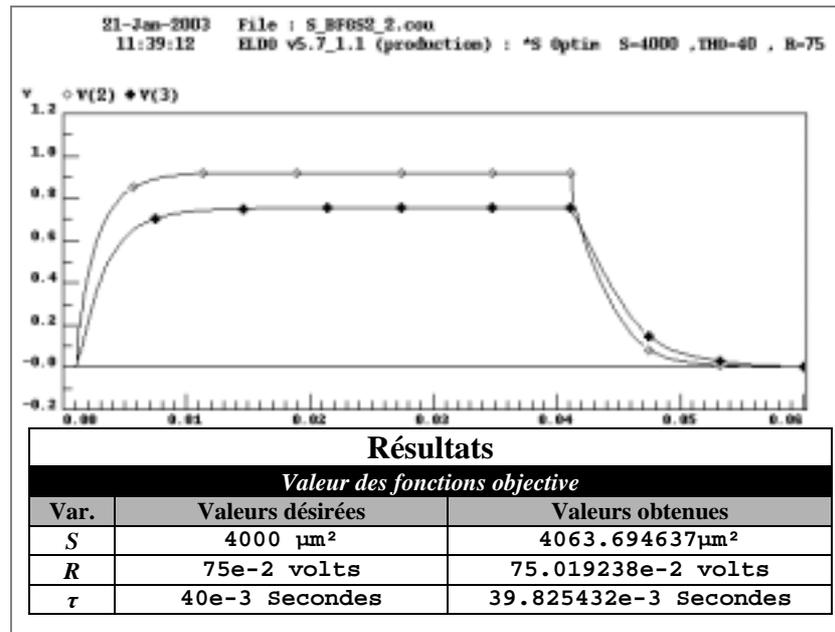


Figure 6.35. Résultats de simulation du comportement du convertisseur avant l'optimisation : (courbe $v(2)$) et après l'optimisation : (courbe $v(3)$) (S comme fonction objective principale, l'algorithme BFGS, $H_0 = \nabla^2 f(x_0)$, deuxième point de départ), et valeurs finales des trois fonctions objectives.

La performance obtenue pour ce point de départ est quasi identique par rapport à la performance obtenue pour le premier point de départ, voir les tableaux comparatifs de la Figure 6.33 et de la Figure 6.35. En terme de minimisation de la surface, on obtient $s^* = 4063.694637 \mu\text{m}^2$ contre $s^* = 4065.668255 \mu\text{m}^2$ pour le premier point de départ. Les amplitudes de la réponse sont très proches ($r^* = 75.019238e-2$ volts contre $r^* = 74.995423e-2$ volts pour le premier point de départ), et finalement les deux valeurs de constante de temps sont quasi identique ($\tau^* = 39.825432e-3$ secondes contre $\tau^* = 39.991212e-3$ secondes pour le premier point de départ).

IV. Comparaison des deux méthodes

Même si l'on arrive à avoir des performances satisfaisantes avec les deux algorithmes (avec une supériorité pour l'algorithme BFGS $H_0 = \nabla^2 f(x_0)$ au niveau de la constante de temps), le problème de la convergence de l'algorithme BFGS pèse lourdement sur son efficacité dans ce mode d'optimisation. Nous pensons que la méthode BFGS $H_0 = \nabla^2 f(x_0)$ est la mieux adaptée pour ce genre de problème d'optimisation pour deux raisons principales :

- La stabilité de la convergence par rapport au choix de point de départ.
- La convergence sûre pour des domaines de définition assez larges.

6.4.2.4 Comparaison entre les trois modes d'optimisation

Nous avons formulé le problème général de l'optimisation du convertisseur électrothermique en trois modes d'optimisation. Ces modes correspondent à :

1. τ comme fonction objective
2. R comme fonction objective
3. S comme fonction objective

Pour chacun de ces modes nous avons appliqué différents algorithmes et analysé les différents résultats de l'optimisation pour trouver l'algorithme le mieux adapté au mode d'optimisation utilisé. Notre analyse se basait sur deux critères essentiels :

- Les différents paramètres de convergence d'algorithme : est-ce que l'algorithme converge ? quelle est sa vitesse de convergence ? quels sont les domaines de convergence, etc.
- La précision de la performance obtenue par l'algorithme par rapport à la performance désirée, valeurs finales de τ , S et R par rapport aux valeurs désirées.

Pour chaque mode, nous avons élu l'algorithme le mieux adapté à ce mode : le gradient de la plus forte pente descendante ou le BFGS $H_0=I$ pour le mode où τ est la fonction objective, et le BFGS $H_0 = \nabla^2 f(x_0)$ pour les deux autres modes.

Pour finir, il nous reste à définir les critères du choix du mode d'optimisation, en d'autres termes étant donné une performance désirée, définie par les fonctions τ , S , et R , quelle fonction parmi ces trois fonctions doit-on choisir comme fonction objective principale ?

La réponse à cette question dépend du degré de liberté de chacune de ces fonctions, d'une manière générale l'algorithme BFGS $H_0 = \nabla^2 f(x_0)$ est le plus stable parmi les trois algorithmes étudiés et il convient aux trois modes d'optimisation. L'inconvénient de cet algorithme ne réside pas dans son exécution mais dans son implémentation et sa calibration.

Il y a quelques astuces qui permettent de choisir le mode d'optimisation :

- Transformer la fonction S en contrainte tant que cela est possible : cela permet d'éviter les phénomènes de zigzag et permet le recours à l'algorithme du gradient de la plus forte pente descendante qui est facile à implémenter.
- Dans le cas où l'on connaît à l'avance le nombre de thermocouples N , ou si ce nombre est imposé (paramètre figé), passer en mode de τ comme fonction objective.
- Essayer si possible de limiter le nombre de boucles sur le nombre de thermocouples N , cela permet d'éviter : soit des cas de convergence, soit des itérations inutiles. Il faut toujours se rappeler qu'un algorithme diverge signifie un temps de calcul infini.

6.5 Conclusion

Dans ce chapitre, nous avons montré une méthodologie de synthèse et d'optimisation pour microsystemes permettant au concepteur de créer des composants microsystemes à partir d'une description à haut niveau d'abstraction. L'idée était de partir d'une description comportementale d'un composant microsysteme et de générer son dessin de masques en interprétant cette description. Il a fallu résoudre plusieurs problèmes :

1. La limitation des langages comportementaux à fournir des informations sur la topologie de composant microsysteme décrit.
2. L'absence de bibliothèque standard microsystemes dans le flot de conception.
3. La difficulté à définir le problème d'optimisation des microsystemes et ensuite à choisir la technique adéquate d'optimisation.

Un langage hybride de description de microsystemes a été introduit. Ce langage convient mieux que les langages comportementaux de type HDL-A, puisqu'il combine la description géométrique et la description comportementale. Des composants microsysteme de base ont été définis par ce langage ce qui nous permet de construire une bibliothèque pour le composant microsysteme utilisé.

En ce qui concerne la partie optimisation, nous avons exposé plusieurs méthodes. Nous avons appliqué ces méthodes sur le convertisseur électrothermique en imposant une performance souhaitée. Nous avons qualifié l'efficacité de ces méthodes selon leur convergence (ou divergence) vers la performance désirée et selon la précision de cette performance par rapport à la performance obtenue.

Le choix du convertisseur électrothermique comme application, pour la méthodologie de synthèse et pour l'optimisation a été fait pour deux raisons essentielles : premièrement, le convertisseur électrothermique est un composant microsysteme qui a été bien étudié, modélisé et fabriqué dans le groupe microsysteme de TIMA, deuxièmement l'optimisation de ce composant engage plusieurs variables de natures différentes telles que la réponse thermique comme variable d'énergie, la constante de temps comme variable de temps et la surface comme variable de géométrie. La méthodologie d'optimisation que l'on a appliqué sur ce composant peut être facilement étendue vers d'autres composants microsystemes.

Néanmoins, il y a des améliorations possibles à la méthodologie d'optimisation définie dans ce chapitre. On peut améliorer la vitesse de convergence, en utilisant des combinaisons d'algorithmes d'optimisation plutôt que d'en utiliser qu'un seul. Cela peut se faire en fractionnant les domaines de définition des paramètres dans des régions de confiance pour chaque algorithme. Une région de confiance est la partie de l'espace de conception, espace multidimensionnel défini par les paramètres de conception, où l'algorithme converge avec une vitesse raisonnable et un temps de calcul satisfaisant.

Références de Chapitre 6

- [1] K. R. Laker, W. M. C. Sansen, "Design of Analog Integrated Circuits and Systems". Mac Graw Hill, 1994.
- [2] J. Rijmenants, J. B. Litisios, T. R. Schwarz and M. G. R. Degrauwe, "ILAC: Automated Layout tool for Analog CMOS Circuits". *IEEE Journal of Solid-State Circuits*, vol. 24, n° 24, pp. 417-425, April 1989.
- [3] M. Kayal, S. Piguet, M. Declercq and B. Hochet, "SALIM: A Layout Generation Tool for Analog ICs", *IEEE Journal of Solid-State Circuits*, pp. 751-754, May 1988.
- [4] J. M. Cohn, D. J. Garrod, R. A. Rutenbar, L. R. Carley, "Analog device-level Layout Automation", New York: Kluwer Academic, 1994.
- [5] E.K. Antonsson, "Structured Design Methods for MEMS", *NSF sponsored workshop on Structured Design Methods for MEMS*, November 12-15, 1995.
- [6] Z. Juneidi, K. Torki, B. Charlot, B. Courtois, "MEMS Synthesis and Optimization", *Design, Test, Integration and packaging of MEMS/MOEMS*, Cannes, France, April 2001.
- [7] T. Mukherjee and G. K. Fedder, "Structured Design Of Microelectromechanical Systems", *Proceedings of the 34th Design Automation Conference (DAC '97)*, Anaheim, CA, June 9-13, 1997.
- [8] G. K. Fedder, "Structured Design of Integrated MEMS", *Proc. of MEMS '99*, Orlando, Florida, pp.1-8, Jan. 17-21, 1999.
- [9] T.J. Wagner, "Hierarchical Layout Verification", *IEEE Design and Test*, pp. 31-37, February 1985.
- [10] J.F. Bonnans, J.C. Gilbert, C. Lemarechal, C. Sagastizabal, "Optimisation numérique aspects théoriques et pratiques", Springer, ISBN 3-540-63183-6, 1997.
- [11] L. Polak, "Optimization Algorithms and Consistent Approximations", Springer, ISBN 0-387-94971-2, 1997
- [12] P. E. Gill, W. Murray and M.H. Wright, "Practical Optimization", ACADEMIC PRESS ISBN 0-12-283952-8, 1999 (second edition).

7

Conclusions et perspectives

Cette thèse porte sur trois axes complémentaires dans le but de fournir un environnement CAO plus complet pour microsystemes. Ces axes sont les suivants :

- Des techniques pour améliorer la qualité de dessin des masques des microsystemes et la vérification de règles de dessin : les principaux problèmes que nous avons affrontés dans ce domaine étaient l'usage intensif des formes Non-Manhattan dans la conception de microsystemes qui posait le problème d'échantillonnage de ces formes et la compatibilité avec les algorithmes de vérification de règles de dessin.
- Une méthodologie de synthèse et d'optimisation pour microsystemes permettant au concepteur de créer des composants microsystemes à partir d'une description à haut niveau d'abstraction : l'idée était de partir d'une description comportementale d'un composant microsysteme et de générer son dessin de masque en interprétant cette description. Il a fallu résoudre plusieurs problèmes :
 1. La limitation de langages comportementaux à fournir des informations sur la topologie de composant microsysteme décrit.
 2. L'absence de bibliothèque standard microsystemes dans le flot de conception.
 3. La difficulté de définir le problème d'optimisation des microsystemes et ensuite de choisir la technique adéquate d'optimisation.
- Un environnement de simulation global de microsystemes : dans cet environnement des composants microsystemes sont simulés avec leur environnement électronique et jusqu'au niveau du boîtier. Des telles simulations impliquent deux tâches :
 - La modélisation des différents composants de différents domaines d'énergie mécanique/thermique/analogique/numérique.

- L'intégration des différents simulateurs dans un seul environnement de simulation globale.

Les travaux pendant cette thèse nous ont permis :

- D'implémenter des nouvelles techniques de vérification de règles de dessin implémentées et de les introduire dans les trois kits de conception des fonderies Bosch, AMS, et SensoNor dans le cadre du projet européen « ESPRIT 28124 TALENT ». Ces techniques utilisées par les concepteurs des trois fonderies ont permis de vérifier le layout de microsystèmes avec efficacité et sans introduire de pénalités de performance.
- De démontrer une méthodologie de synthèse et d'optimisation. Un langage hybride de description de microsystèmes a été introduit. Ce langage convient mieux que les langages comportementaux de type HDL-A, puisqu'il combine la description géométrique et la description comportementale. Des composants microsytème de base ont été définis par ce langage ce qui nous permet de construire une bibliothèque pour le composant microsytème utilisé. Différents algorithmes d'optimisation ont été discutés. Une application de cette méthodologie sur un convertisseur électrothermique a été exposée.
- De définir une méthodologie de simulation globale de *System On Chip*. Cette méthodologie a été validée sur la co-simulation d'un commutateur optique. Dans la co-simulation réalisée, nous avons couplé un module électromécanique spécifié sur MATLAB avec plusieurs modules optiques spécifiés sur C++ et un module de commande spécifié directement sur *SystemC*. Nous avons appliqué la co-simulation pour réaliser une première validation du système à très haut niveau. Par ailleurs, cette méthode de validation peut s'appliquer tout au long du flot de conception au fur et à mesure que d'autres modules sont incorporés au système ou que certains modules sont raffinés.

En ce qui concerne les éditeurs de dessin de masques de microsystèmes, nous pensons qu'une nouvelle génération de ces outils est nécessaire. Dans de tels éditeurs de layout, les formes géométriques seront représentées par leurs équations analytiques et non pas par des polygones à grands nombres de sommets. Cette approche sera beaucoup plus efficace encore, et ouvre des perspectives nouvelles pour l'implémentation des outils agissant sur la géométrie, en traitant les objets comme des entités compactes et non discrétisées.

Nous pensons que la méthodologie de synthèse et d'optimisation pour microsystèmes que nous avons développé fera le cadre général des futurs travaux. Il nous reste à enrichir le langage de synthèse que nous avons défini d'une manière à ce que la phase d'optimisation soit moins coûteuse.

Dans le domaine de co-simulation, les outils de la CAO pour les microsystèmes doivent évoluer progressivement afin de permettre la modélisation de systèmes non seulement de domaines multiples mais de niveaux d'abstraction multiples. Ainsi, ces outils faciliteront la création de modèles exécutables qui seront vus comme des prototypes virtuels. Une fois les systèmes validés par simulation de manière exhaustive, il sera donc possible de procéder avec moins de risques à une simulation plus fine et à la fabrication de prototypes physiques.

Publications

- [1] Z. Juneidi, K. Torki, R. Hamza, "Design Rules for Non Manhattan Shapes", *SPIE Micromachining and Microfabrication*, Santa Clara, Ca, USA, September 2000.
- [2] Z. Juneidi, K. Torki, B. Charlot, B. Courtois, "MEMS Synthesis and Optimization", *Design, Test, Integration and packaging of MEMS/MOEMS*, Cannes, France, April 2001.
- [3] Z. Juneidi, K. Torki, G. Nicolescu, S. Martinez, B. Courtois, A. A. Jerraya, "Global modeling and simulation of systems on chip embedding MEMS devices", *ASICON 2001*, shanghi, China, 2001.

Annexe A : Modèle HDL-A d'un convertisseur électrothermique

```

--modele de convertisseur electrothermique en poutre avec resistance au bout--

entity ETC3 is
generic
(
    z: integer; --nbr de thermocouples--
    alpha1, alpha2: analog; --coef seebeck--
    tcrpoly: analog; --coef de variation de resistance avec la
        temperature--
    Rsqpoly: analog; --resistance carre de poly--
    Rsqmet: analog; --resistance carre du metal--
    gmet: analog; --conduction thermique du metal--
    gpoly: analog;
    cpoly: analog; --capacite thermique du poly--
    hair: analog; --convection thermique de l'air--
    emit: analog; --emittance--
    sb: analog; --constante de stephan boltzmann--
    gox: analog; --conductance thermique de l oxide--
    cox: analog; --capacite thermique de l oxide--
    gpass: analog;
    cpass: analog;
    T0: analog; --temperature ambiante--
    elocos: analog; --epaisseurs--
    epoly1: analog;
    epoly2: analog;
    eox1: analog;
    eox2: analog;
    eox3: analog;
    epass: analog;
    emet1: analog;
    emet2: analog;
    lonptr: analog; --longueurs,largeurs--
    larptr: analog;
    lontp: analog;
    lonox: analog;
    larpoly1: analog;
    larpoly2: analog;
    larmet: analog;
    lonres: analog;
    larres: analog;
    aox: analog;
    atp: analog;
    rfault: analog
);

pin(
    vinp,vinn: electrical; --entrees--
    voutp,voutn: electrical; --sorties--
    Tth,Toxi,T1,T2,T3,T4,T5,T6: electrical --temperatures--
);

end entity ETC3;
architecture b of ETC3 is

state vin,i0,v0: analog;
state power: analog;
state theater: analog;
state tox: analog;

```

```

state tp1: analog;
state tp2: analog;
state tp3: analog;
state tp4: analog;
state tp5: analog;
state tp6: analog;
state rtp: analog;
state rthox: analog;
state cthox: analog;
state rthtp: analog;
state ctthp: analog;
state cthres: analog;
state rthlig: analog;
state S1: analog;
state Sox: analog;
state Sth: analog;
state nth: analog;

begin
  relation
  procedural for init =>

    z:=16;
    alpha1:=248.7e-6;
    alpha2:=0.0e-6;
    tcrpoly:=8.8e-4; --K-1 --
    Rsgpoly:=25.0; --ohm --
    Rsgmet:=0.08; --ohm --
    gmet:=238.0; --W/m/K --
    gpoly:=28.0; --W/m/K --
    cpoly:=1.6e6; --J/m3/K --
    hair:=479.0; --W/m2/K--
    emit:=1.0;
    sb:=5.67e-8; --W/m2/K-4--
    gox:=1.25; --W/m/K --
    cox:=1.05e6; --J/m3/K --
    gpass:=30.0;
    cpass:=1.0e6;
    T0:=300.0; --K --
    elocos:=0.4e-6; -- metres --
    epoly1:=0.25e-6;
    epoly2:=0.25e-6;
    eox1:=0.04e-6;
    eox2:=0.55e-6;
    eox3:=0.75e-6;
    epass:=1.2e-6;
    emet1:=0.72e-6;
    emet2:=0.94e-6;
    lonptr:=265.0e-6; -- metres --
    larptr:=120.0e-6;
    lontp:=255.0e-6;
    lonox:=13.0e-6;
    larpoly1:=0.8e-6;
    larpoly2:=1.6e-6;
    larmet:=1.4e-6;
    lonres:=105.0e-6;
    larres:=2.5e-6;
    aox:=0.5;
    atp:=0.5;
    rfault:=1.0e10;
    nth:=3.0;
    rthox:=gox*larptr*(elocos+eox1+eox2+eox3)/lonox gpass*larptr*(epass)/lonox
    +gmet*larmet*emet1/lonox*2.0;

    cthox:=cox*larptr*(elocos+eox1+eox2+eox3)*lonox;
    Tth.v%=theater;
    Toxi.v%=tox;
    T1.v%=tp1;
    T2.v%=tp2;
    T3.v%=tp3;

```

```

T4.v%=tp4;
T5.v%=tp5;
T6.v%=tp6;
v0:=(tp1-T0)*real(z)*(alpha1-alpha2)+i0*rtp;
rtp:=real(z)*rsqpoly*(lonp/larpoly1+lonp/larpoly2)*(1.0+tcrcpoly*(tp2-T0));

equation(theater,tox,tp1,tp2,tp3,tp4,tp5,tp6) for
ac,transient=>vin*vin/(rsqpoly*lonres/larres*(1.0+tcrcpoly*(theater-
T0)))=aox*(rthox)*(theater-tox);

aox*rthox*(theater-tox)==cthox*tox'ddt+(1.0-aox)*(rthox)*(tox-tp1)+(tox-
T0)*Sox*hair+4.0*emit*Sox*sb*T0*T0*T0*(tox-T0);

(1.0-aox)*(rthox)*(tox-tp1)==atp*(rthtp)*(tp1-tp2);

atp*(rthtp)*(tp1-tp2)==cthtp*tp2'ddt+(1.0-atp)*(rthtp)*(tp2-tp3)
+(tp2-T0)*Sth*hair+4.0*emit*Sth*sb*T0*T0*T0*(tp2-T0);(1.0-atp)*(rthtp)*(tp2-
tp3)==atp*(rthtp)*(tp3-tp4);

atp*(rthtp)*(tp3-tp4)==cthtp*tp4'ddt+(1.0-atp)*(rthtp)*(tp4-T0)+(tp4-
T0)*Sth*hair+4.0*emit*Sth*sb*T0*T0*T0*(tp4-T0);

(1.0-atp)*(rthtp)*(tp4-tp5)==atp*(rthtp)*(tp5-tp6)+rfault*(tp5-T0);

atp*(rthtp)*(tp5-tp6)==cthtp*tp6'ddt+(1.0-atp)*(rthtp)*(tp6-T0)+(tp6-
T0)*Sth*hair+4.0*emit*Sth*sb*T0*T0*T0*(tp6-T0);

end relation;

end architecture b ;

```

Annexe B: Bosch technology design rules

1. Description of Technology

1.1. General

The Bosch process features a polycrystalline silicon layer with a thickness of 10.3 μm for the freestanding structures. The large thickness is achieved using an epitaxial deposition of polysilicon ('epipoly'). A special trench technique allows the formation of vertical sidewalls with high aspect ratio. In addition to the functional polysilicon layer a second thin poly-Si-layer ('buried poly') is provided underneath, which serves as interconnect, shield or counter-electrode. The buried poly layer is isolated from the epipoly by the sacrificial oxide and from the substrate by a lower oxide.

1.2. Process Flow

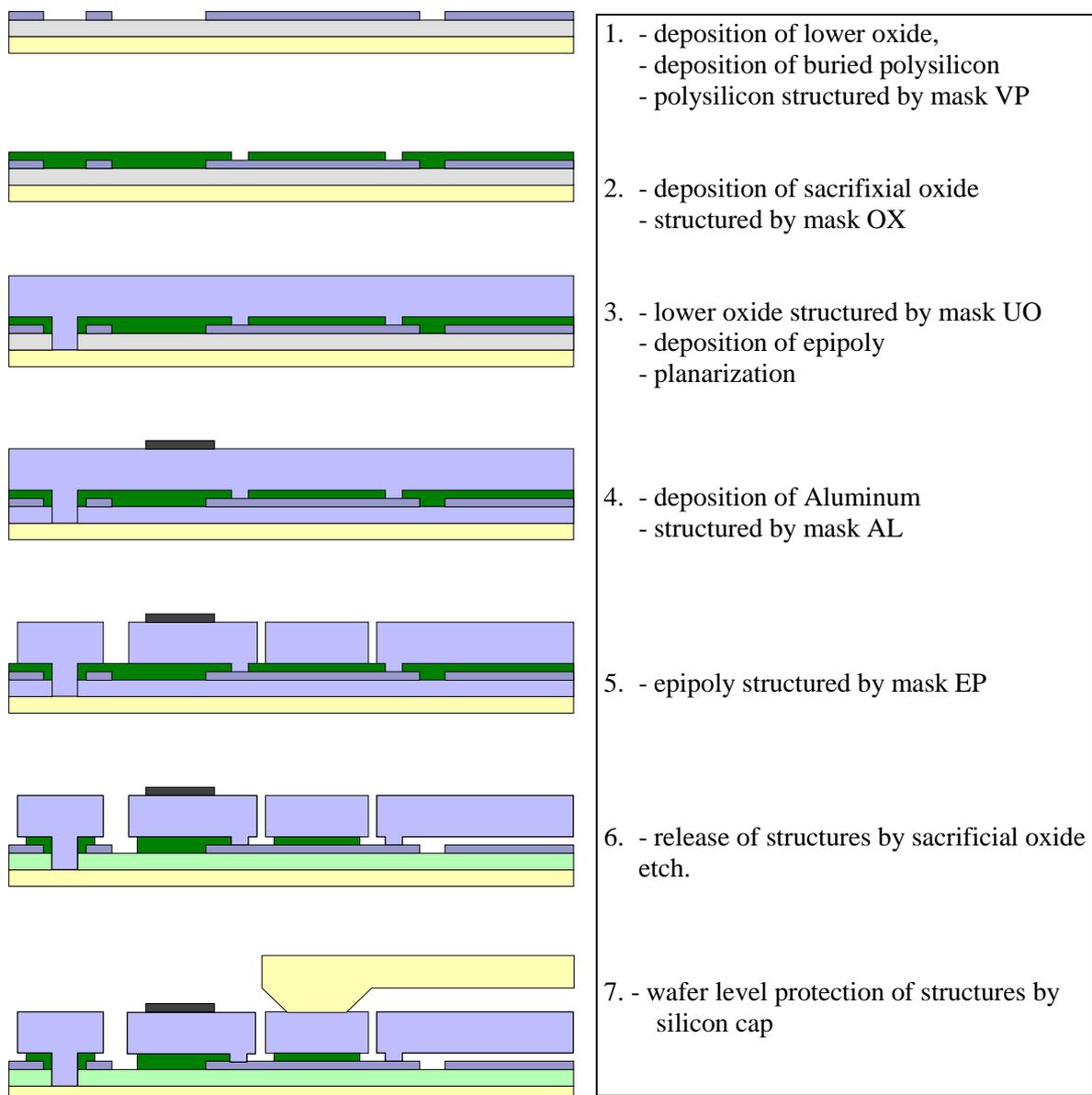


Fig. 1.1: Technology flow

2. Mask design rules

(All dimensions in micrometer except stated otherwise)

2.1 Color Code

Mask	Name	Type	Function	Color Code
1	VP	Dark	Definition of buried polysilicon structure for interconnects, shields, bottom electrodes, structural anchors	
2	OX	Clear	Definition of contact holes. Connection of epipoly layer to buried poly layer.	
3	UO	Clear	Definition fo contact areas to substrate. Connects epipoly layer to substrate	
4	AL	Dark	Definition of Aluminum interconnects and bond pads	
5	EP	Dark	Definition of epipoly structures	

Dark mask: remaining areas are digitized (e.g. electrical connect in VP)
Clear mask: removed areas are digitized (e.g. contact holes in OX)

2.2 Diagrams

The following diagrams are intended to illustrate generic structure elements of the micromachining technology. The dimensions are given in the following sections.

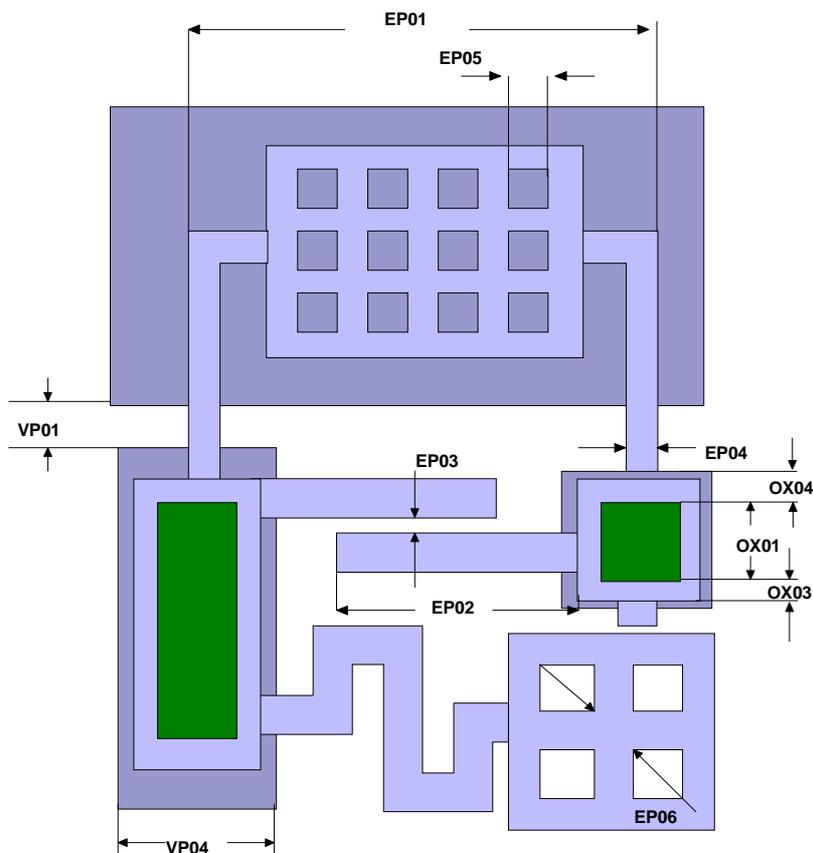


Fig. 2.1.: Generic structure, illustrating major surface micromachining designrules.

2.3 Mask VP

This mask structures the buried polysilicon layer. It is often used as a buried interconnect layer, e.g. to connect the functional structures with the bond pads. It may also be used to define electrodes, e.g. when capacitive detection of vertical movement of a Epipoly mass is required.

This mask is a dark-tone mask, i.e. buried polysilicon areas are digitized.

Physical Dimension = Layout - 1.0 μm

Rule	Description	Layout
VP01	Distance between Buried Poly Areas	> 4.0
VP02	-----	---
VP03	Width (VP covered by EP, overlap ≥ 10.0)	> 10.0
VP04	Width (EP overlap < 10.0 or no EP above VP))	> 40.0
VP05	Area coverage (buried poly)	> 15% < 30%

2.4 Mask OX

This mask structures the sacrificial oxide layer. It is used to provide electrical connection between buried polysilicon and epitaxial polysilicon. Also mechanically anchors epitaxial polysilicon to the buried polysilicon.

This mask is a clear-tone mask, i.e. areas where oxide is removed are digitized.

Physical Dimension = Layout + 0.4 μm

Rule	Description	Layout
OX01	Contact	> 4.0 < 25.0
OX02	Distance between contacts	> 4.0
OX03	Enclosure by EP	> 3.5
OX04	Enclosure by VP	> 3.5
OX05	Area coverage (holes in oxide)	> 7% < 15%

2.5 Mask UO

This mask provides the opportunity to create a contact to the substrate by removing both oxide layers before growth of epitaxial silicon. It should only be used in the bond pad area of the chip.

This mask is a clear tone mask, i.e. the area where oxides are removed are digitized

Physical Dimension = Layout + 24 μm

Rule	Description	Layout
UO01	Contact to substrate	> 50
UO02	Distance to VP	> 15
UO03	Enclosure by EP	> 10

2.6 Mask AL

This mask structures Aluminum. It is primarily used to define bond pads.
This mask is a dark-tone mask, i.e. aluminum areas are digitized.

Physical Dimension = Layout - 3.0 μm

Rule	Description	Layout
AL01	Width	> 4.0
AL02	Enclosure by EP	> 3.5

2.7 Mask EP

This is the essential mask for micromachined structures. It defines the dimensions of the epitaxial polysilicon.
This mask is a dark-tone mask, i.e. epipoly areas are digitized.

Physical Dimension = Layout - (0.7 μm +/- 0.5 μm)

Rule	Description	Layout
EP01	Free structure (fixed on both sides)	< 1000
EP02	Free structure (fixed on one side)	< 400
EP03	Spacing between EP structures	> 2.0
EP04	Width	> 3.0
EP05	Etch hole	> 4.0
EP06	Width for structures that are to be released after sacrificial oxide etch	< 5.8
EP07	---	---
EP08	Spacing to structures smaller than 4.0	> 1.5
EP09	Spacing around bond pads	> 50
EP10	Area coverage (polysilicon)	< 94%

3. Bosch design rules in Mentor Graphics Platform

```

////////////////////////////////////
//
// File      : bosch.rules
//
// Purpose   : DRC Rule File for bosch technology
//
// Update    : 19/01/98
//
////////////////////////////////////

TITLE "DRC Rules File for BOSCH Mikroelektronik process"

LVS COMPONENT TYPE PROPERTY comp

// Layer definitions

layer vp      9
layer ox     26
layer uo     33
layer al      7
layer ep      8
layer bp      1
layer fo      2
layer fb      3
layer Chip_Area 80
layer Layout_Area 81
layer holeEP  97

// Merging layers for DRC checking

dvp = OR vp
dox = OR ox
duo = OR uo
dal = OR al
dep = OR ep
dfo = OR fo
dfb = OR fb
dbp = OR bp
dca = OR Chip_Area
dla = OR Layout_Area
dhpep = OR holeEP
eh = HOLEs dep

////////////////////////////////////
//
//                               Design Rules
//
////////////////////////////////////

PRECISION 1000
RESOLUTION 100

DRC TOLERANCE FACTOR 0.0707

////////////////////////////////////
//
// MASK VP, DIFINITION OF BURIED POLYSILICON STRUCTURE
//
////////////////////////////////////
VP01
{
  @ Spacing between buried poly Areas must be > 4.0
  EXTERNAL dvp < 4.0
}

```



```

@ Width of contacts to substrate must be > 50.0 microns
INTERNAL duo < 50.0
}
U002
{
@ Spacing of UO mask to VP mask must be > 15.0 microns
EXTERNAL duo dvp < 15.0
}
U003
{
@ Enclosure of UO mask by EP mask must be > 10.0 microns
ENCLOSURE duo dep < 10.0 ABUT == 0 SINGULAR
}
////////////////////////////////////
//          END MASK UO RULES
////////////////////////////////////

////////////////////////////////////
//          MASK AL RULES
////////////////////////////////////
AL01
{
@ Width of AL structures must be > 4.0 microns
INTERNAL dal < 4.0
}
AL02
{
@ Enclosure of AL mask by EP mask must be > 3.5 microns
ENCLOSURE dal dep < 3.5 ABUT == 0 SINGULAR
}
////////////////////////////////////
//          END MASK AL RULES
////////////////////////////////////
TEMP1 = OR dep dfo
TEMP2 = OR TEMP1 dfb
EP_STRUCT = OR TEMP2 dbp
////////////////////////////////////
//          FREE STRUCTURES RULES
////////////////////////////////////
EP01
{
@ Free structure (fixed on BOTH sides) must be < 1000.0 microns
LENGTH dfb > 1000.0
}
EP02
{
@ Free structure (fixed on ONE side) must be < 400.0 microns
LENGTH dfo > 400.0
}
////////////////////////////////////
//          END FREE STRUCTURES RULES
////////////////////////////////////

////////////////////////////////////
//          EP STRUCTURE
////////////////////////////////////
EP03
{
@ Spacing of EP structure to EP structure EQUAL OR BIGGER than 4.0 must be > 2
Z = LENGTH EP_STRUCT > 4.0
EXTERNAL EP_STRUCT Z < 2.0
}
EP04
{
@ Width of Epipoly Structures must be > 3.0 microns
INTERNAL EP_STRUCT < 3.0
}
EP05
{
@ Width of Epipoly Holes must be > 4.0 microns
INTERNAL eh < 4.0
INTERNAL dhep < 4.0
}

```

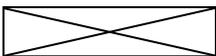
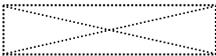
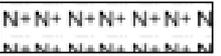
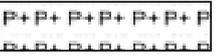
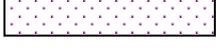
```
EP06
{
  @ Width of Epipoly Structures released after sacrificial oxide etch must be > 5.8 microns
  X = SIZE dhep BY -2.05
  Y = SIZE dhep BY 2.05
  length Y > 4.10
}
EP08
{
  @ Spacing of EP strcture to EP structure SMALLER than 4.0 must be > 1.5

  Y = LENGTH EP_STRUCT < 4.0
  EXTERNAL EP_STRUCT Y < 1.5
}
EP09
{
  @ Spacing of EP strcture to Bond pad Area must be > 50
  EXTERNAL dep dbp < 50.0
  EXTERNAL dfo dbp < 50.0
  EXTERNAL dfb dbp < 50.0
}
EP10
{
  @ Area Coverage(Polysilicon "EP MASK") must be >94%
  x = POLYNET dca
  y = STAMP dep by x
  NET AREA RATIO x y > 1.6383
}
////////////////////////////////////
//          END MASK EP STRUCTURES RULES
////////////////////////////////////

////////////////////////////////////          END          //////////////////////////////////////
////////////////////////////////////
```

Annexe C: AMS technology design rules

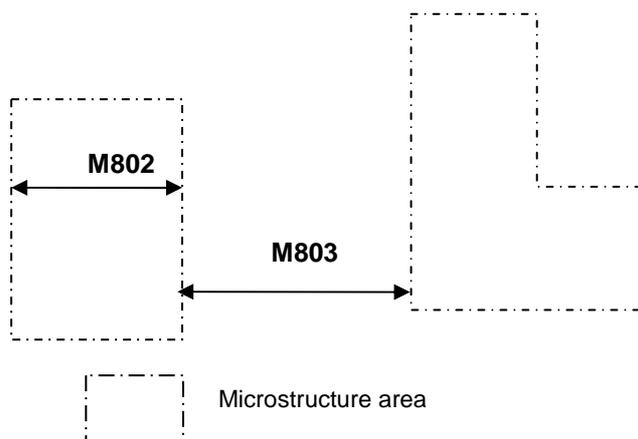
1. Layer symbols

Mask #	TITLE	LAYOUT SYMBOL	NAME
80	MICROSTRUCTURE AREA		MICBULK
18	NWELL		NTUB
22	NWELL IMPLANT		FIMP
20	DIFFUSION / ACTIVE AREA		DIFF
24	POLYSILICON1		POLY1
41	POLYSILICON2		POLY2
26	N+ IMPLANT		NPLUS
28	P+ IMPLANT		PPLUS
30	CONTACT		CONT
32	METAL 1		MET1
34	VIA		VIA
36	METAL 2		MET2
38	PASSIVATION OPENING		PAD

2. Topological design rules

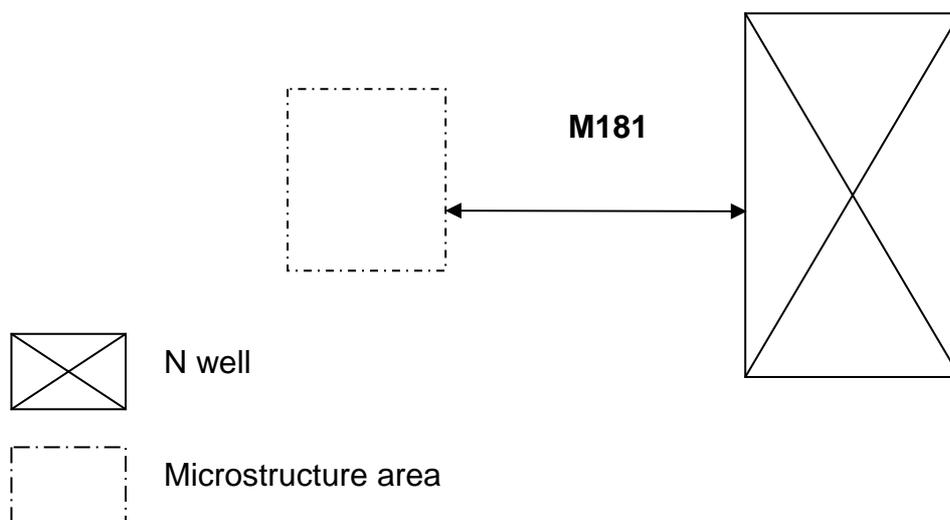
2.1. Layer: MICROSTRUCTURE AREA (MICBULK).

Rule Number	Parameter	min. dimension
M801	edges of Microstructure Area must be parallel to the axes of the CAD software.	
M802	width of Microstructure Area	30
M803	spacing between two Microstructure Areas	35



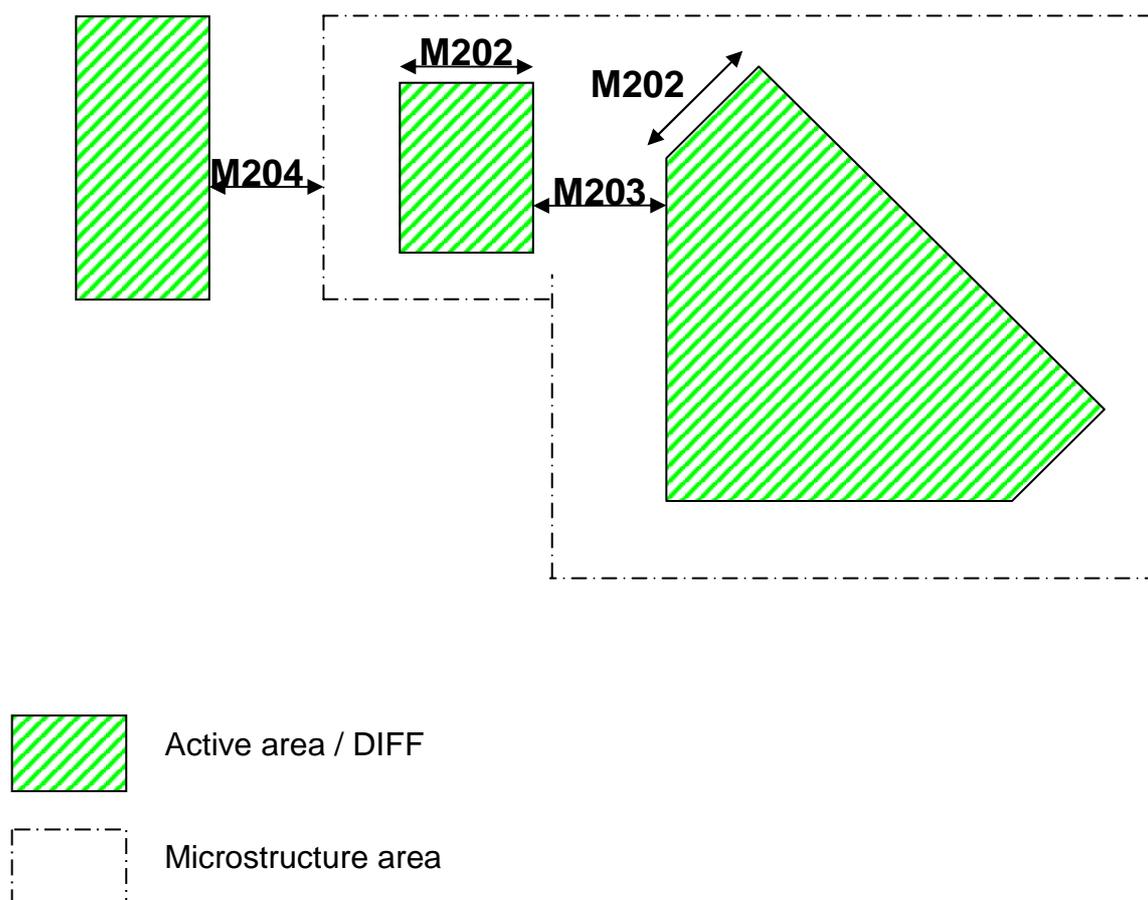
2.2. Layer: N-WELL IMPLANT (NTUB).

Rule Number	Parameter	min. dimension
M181	N-Well spacing to Microstructure Area.	
M182	no coincidence between N-Well and Microstructure Area.	30
M221	no coincidence between FIMP and Microstructure Area.	35



2.3. Layer: DIFFUSION / ACTIVE AREA (DIFF)

Rule Number	Parameter	min. dimension
M201	The vertices of Diffusion shapes must be parallel or make a 45° angle with the axes of the CAD software.	20
M202	width of Diffusion	
M203	spacing between 2 Diffusion	10
M204	spacing to Microstructure Area	35
M205	Diffusion inside Microstructure Area, must be covered with Contact.	
M206	Diffusion, inside Microstructure Area, must be covered with Open Via	
M207	Diffusion, inside Microstructure Area, must enclose Passivation Opening	
M208	no coincidence between Contact and Microstructure Area.	

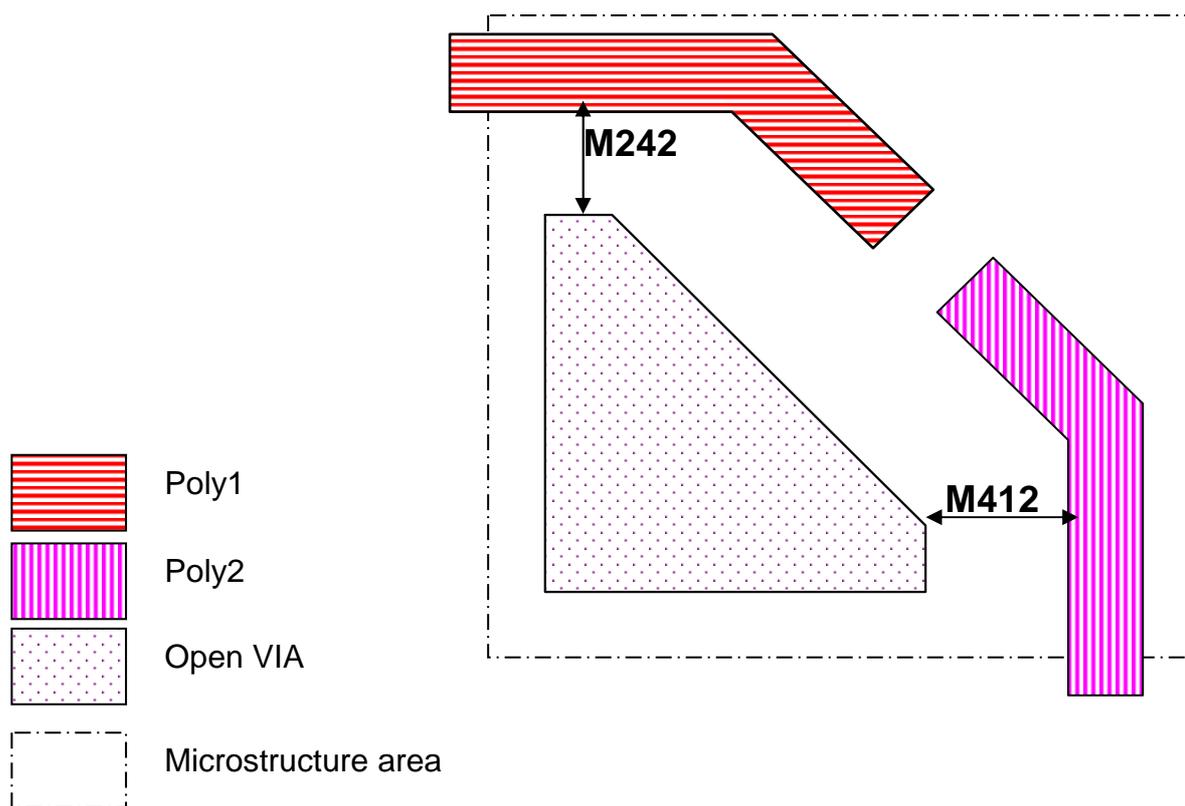


2.4. Layer: POLYSILICON (POLY1)

Rule Number	Parameter	min. dimension
M241	Polysilicon 1 shapes inside Microstructure Areas have to respect all the normal AMS CAE/CYE design rules.	
M242	spacing to open Via	3
M243	no coincidence between Polysilicon and Open Via	

2.5. Layer: POLYSILICON 2 (POLY2)

Rule Number	Parameter	min. dimension
M411	Polysilicon 2 shapes inside Microstructure Areas have to respect all the normal AMS CAE/CYE design rules.	
M412	spacing to open Via	3
M413	no coincidence between Polysilicon 2 and Open Via	

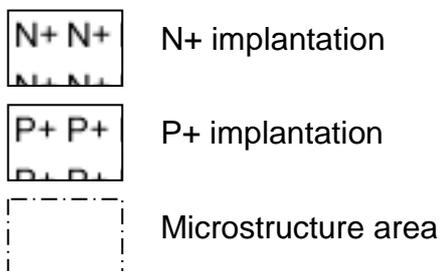
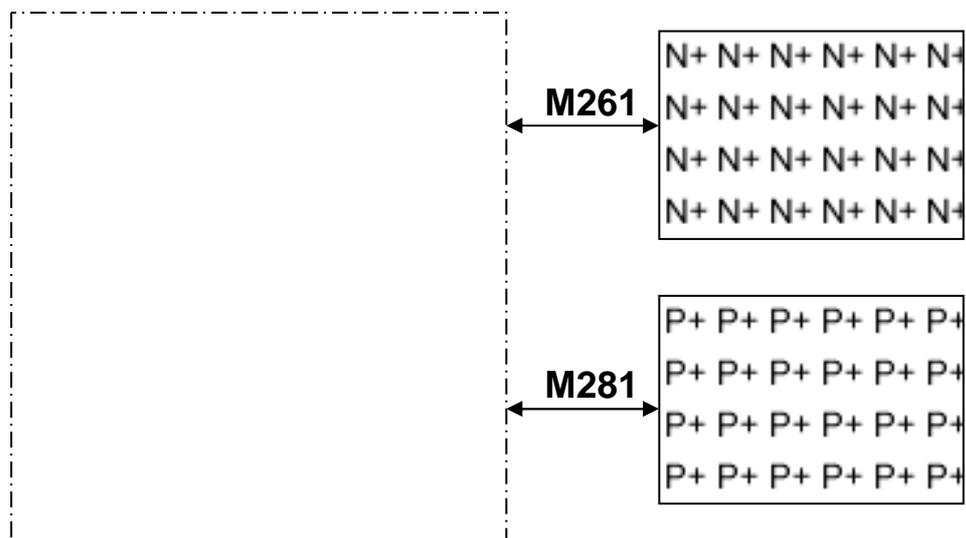


2.6. Layer: N + IMPLANT (NPLUS)

Rule Number	Parameter	min. dimension
M261	N+ implant spacing to Microstructure Area	35
M262	no coincidence between N+ implant and Microstructure Area	

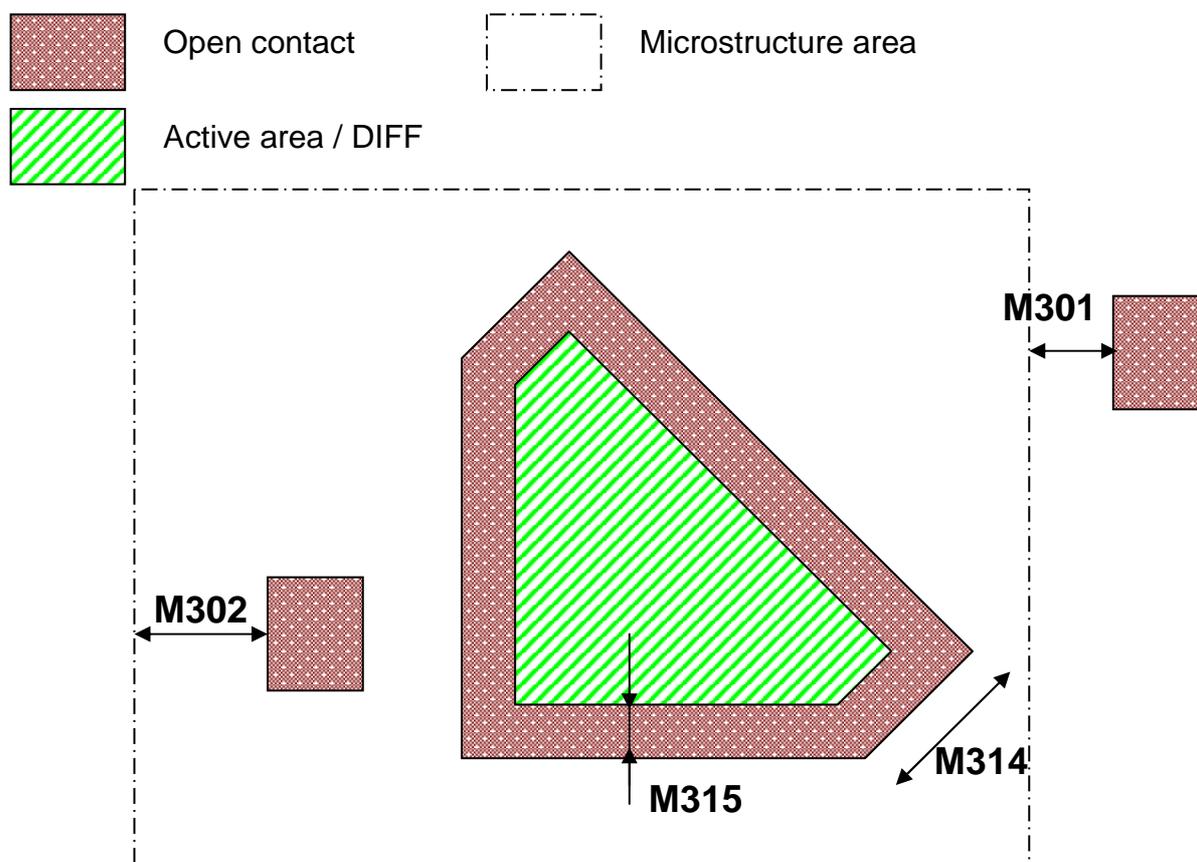
2.7. Layer: P + IMPLANT (PPLUS)

Rule Number	Parameter	min. dimension
M281	P+ implant spacing to Microstructure Area	35
M282	no coincidence between P+ implant and Microstructure Area	



2.8. Layer: OPEN CONTACT/CONTACT (CONT)

Rule Number	Parameter	min. dimension
M300	Contacts inside Microstructure Areas have to respect the AMS standard rules. Open Contacts have to respect the rules of this section	
M301	Contact spacing to Microstructure Area	1
M302	Contact margin to Microstructure Area	1
M303	The vertices of Open Contact shapes must be parallel or make a 45° angle with the axes of the CAD software.	
M304	width of Open Contact	24
M305	overlap of Open Contact to Diffusion	2
M306	Open Contact must enclose Diffusion	
M307	Open Contact must be enclosed inside Open Via	
M308	Open Contact must enclose Passivation Opening	
M309	no coincidence between Contact and Microstructure Area	

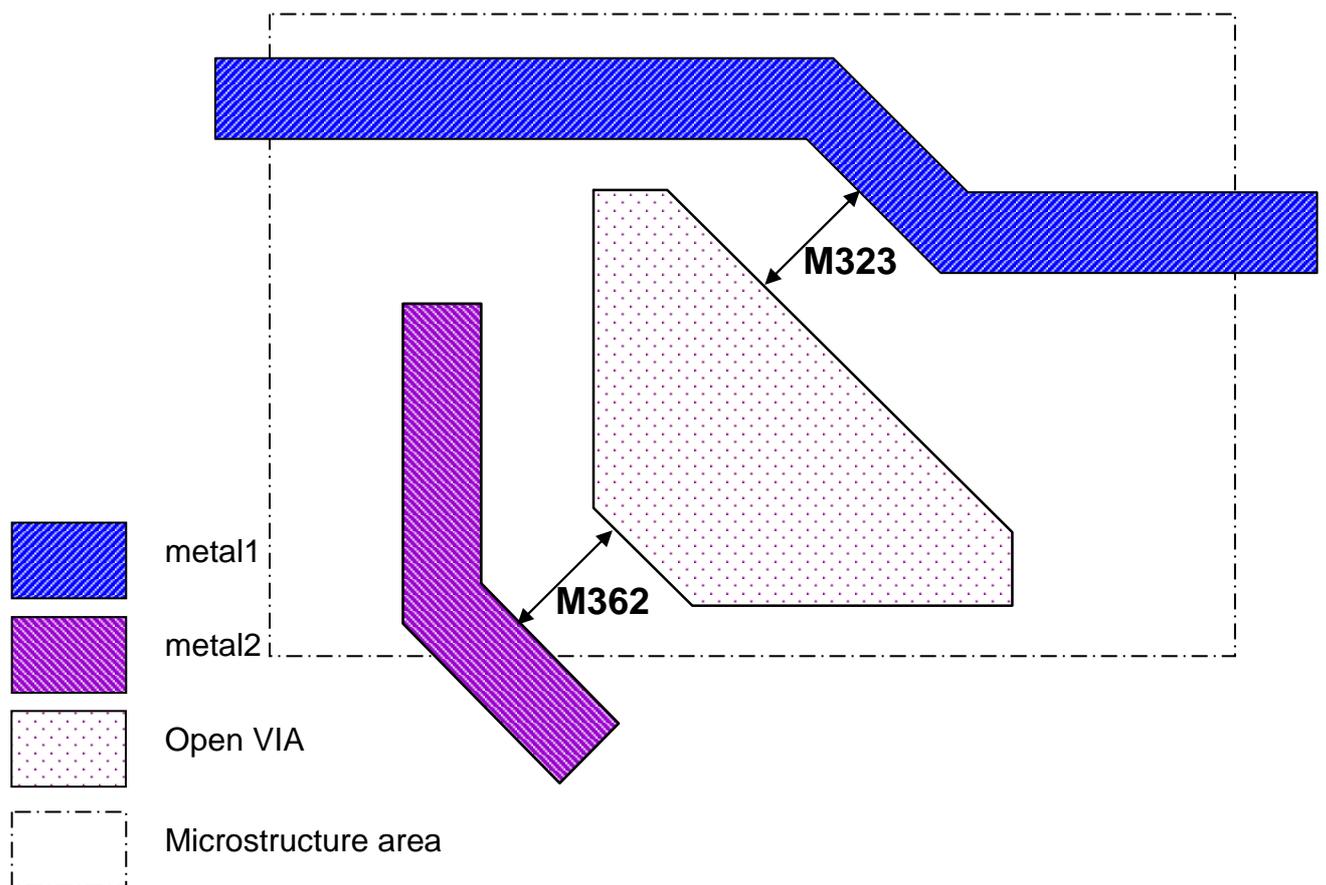


2.9. Layer: METAL 1 (MET1)

Rule Number	Parameter	min. dimension
M321	Metal 1 shapes inside Microstructure Areas have to respect all the normal AMS CAE/CYE design rules	
M322	no coincidence between Metal 1 and open Via	
M323	Metal 1 spacing to open Via	3

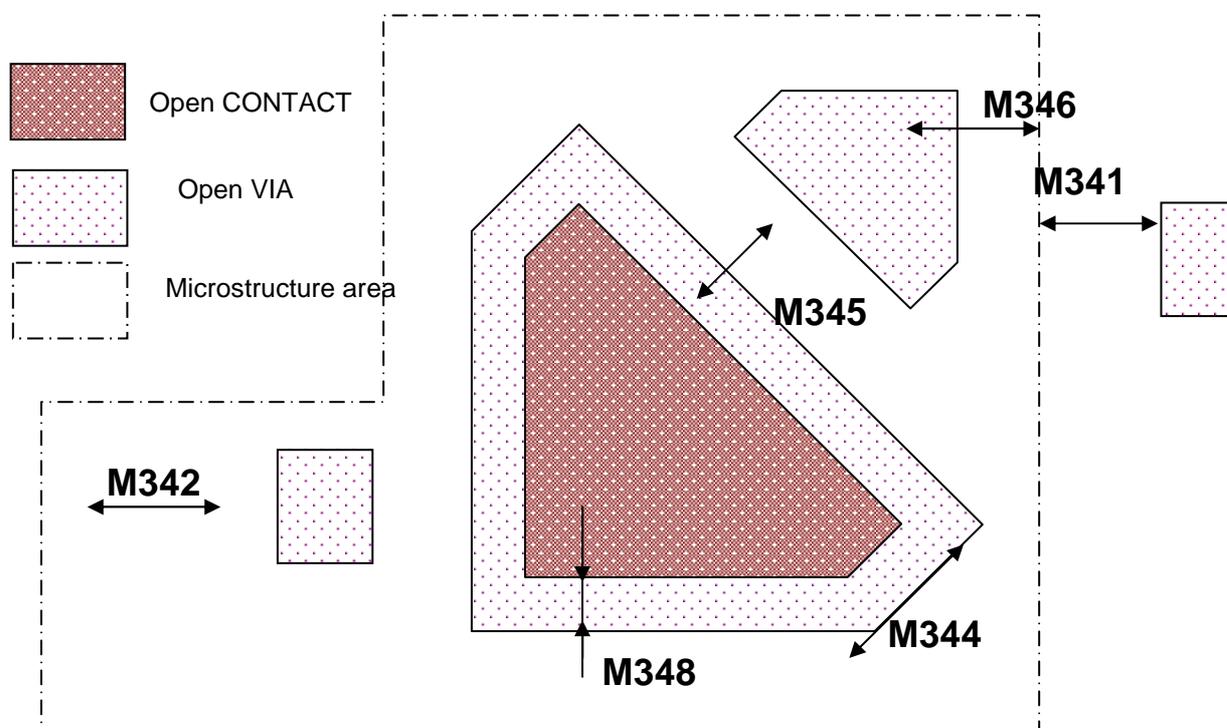
2.10. Layer: METAL 2 (MET2)

Rule Number	Parameter	min. dimension
M361	Metal 2 shapes inside Microstructure Areas have to respect all the normal AMS CAE/CYE design rules, and the following microsystem specific rules.	
M362	Metal 2 spacing to Open Via	3
M363	no coincidence between Metal 2 and Open Via	



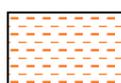
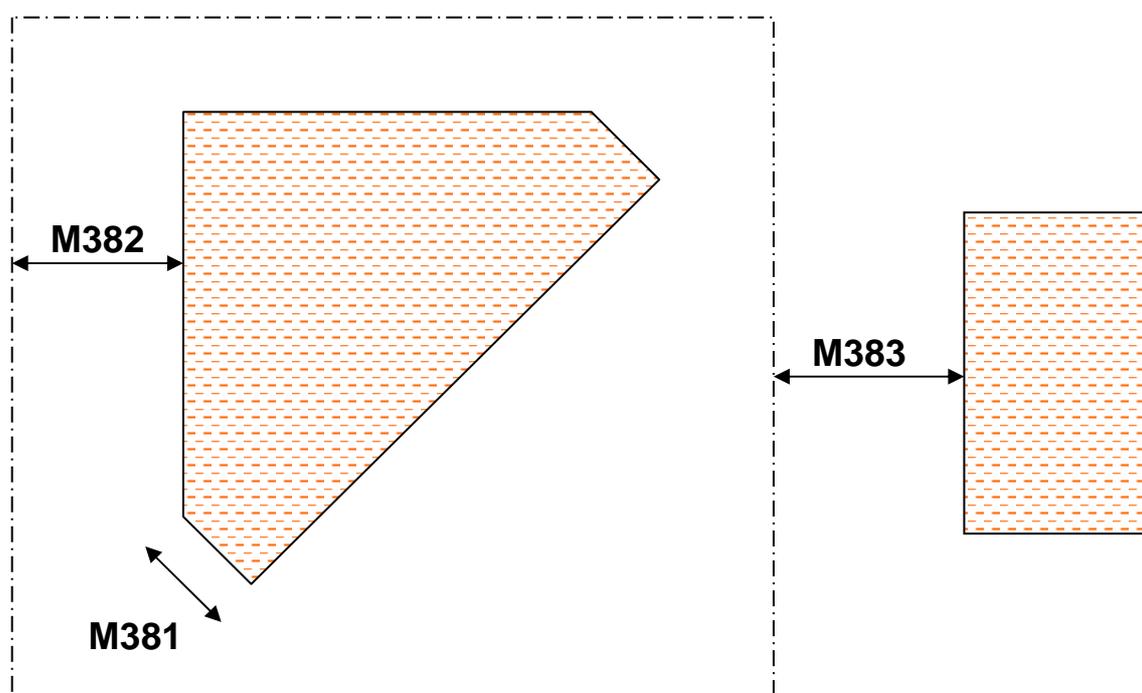
2.11. Layer: OPEN VIA/VIA (VIA)

Rule Number	Parameter	min. dimension
M340	Via Cuts inside Microstructure Areas have to respect the AMS standard rules. Open Vias have to respect the rules presented in this section.	
M341	Metal Via spacing to Microstructure Area	1
M342	Metal Via margin to Microstructure Area	1
M343	The vertices of Open Via shapes must be parallel or make a 45° angle with the axes of the CAD software.	
M344	width of Open Via	28
M345	spacing to Open Via	10
M346	Open Via margin to Microstructure Area	1
M347	Open Via must enclose Diffusion	
M348	overlap of Open Via to Open Contact	2
M349	Open Via must enclose Open Contact	
M3410	Open Via must enclose Passivation Opening.	
M3411	no coincidence between Via and Microstructure area.	



2.12. Layer: PASSIVATION OPENING (PAD)

Rule Number	Parameter	min. dimension
M381	width of Passivation Opening	20
M382	spacing to Microstructure Area	1
M383	margin to Microstructure Area	35
M384	Passivation Opening must be enclosed inside Diffusion Open Contact, Open Via	
M385	The vertices of Passivation Opening shapes must be parallel or make a 45° angle with the axes of the CAD software.	
M386	no coincidence between Passivation Opening and Microstructure area	



PAD WINDOW



Microstructure area

3 AMS 0.8 μ design rules in Mentor Graphics Platform

```

////////////////////////////////////
//
// File      : Ams3.10.drc
//
// Created   : Januaury 99, by Z.JUNEIDI - TIMA MCS Group / CMP
//
// Purpose   : DRC Rule File for AMS 3.10 process
//
// Comments  : Preliminary version
//
// Last Update :02 12 2000
//
////////////////////////////////////

TITLE "DRC Rules File for AMS 3.10 process"

// Layer definitions

layer Open_Area                80
layer N_Well                   1
layer N_Well_Implant           6
layer Diffusion                 4
layer Polysilicon1             10
layer Polysilicon2             14
layer NPLUS_Implant            12
layer PPLUS_Implant            13
layer Contact                  16
layer Metall                   17
layer Via                       18
layer Metal2                   19
layer Passivation_Opening      20

// Merging layers for DRC checking

OPEN= OR Open_Area
NTUB = OR N_Well
FIMP = OR N_Well_Implant
DIFF = OR Diffusion
POLY1 = OR Polysilicon1
POLY2 = OR Polysilicon2
NPLUS = OR NPLUS_Implant
PPLUS = OR PPLUS_Implant
CONT = OR Contact
MET1 = OR Metall
DVIA= OR VIA
MET2 = OR Metal2
PAD = OR Passivation_Opening

////////////////////////////////////
//
//                               Design Rules
//
////////////////////////////////////

FLAG OFFGRID YES
FLAG ACUTE YES
FLAG SKEW YES
PRECISION 1000
RESOLUTION 100

////////// General Rules //////////

offgrid_vertice
{
    DRAWN OFFGRID
}

```

```

acute_angle
{
    DRAWN ACUTE
}

edge_angle
{
    DRAWN SKEW
}

/////////////////////////////////////////////////////////////////
//
//                      Micromachining Design Rules
//
/////////////////////////////////////////////////////////////////

/////////////////////////////////////////////////////////////////
//////////      Open Area (OPEN)          //////////
M801
{
    @ edges of Open Area must be parallel et the axes of CAD software or make 45°
    ANGLE OPEN > 0 < 45
    ANGLE OPEN > 45 < 90
}
M802
{
    @width of Open Area must be >= 30
    INTERNAL OPEN < 30
}
M803
{
    @space between marging of Open Area must be >= 35
    EXTERNAL OPEN < 35 SINGULAR
}
//////////      END Open Area (OPEN)          //////////
/////////////////////////////////////////////////////////////////

/////////////////////////////////////////////////////////////////
//////////      N-WELL + N-WELL IMPLANT (NTUB+FIMP)          //////////

// auxiliary layers //
// The Layer NtubNIOpen contains all the polygons of NTUB which are not inside OPEN //
NtubNIOpen = Not Inside NTUB OPEN
// The Layer FimpNIOpen contains all the polygons of FIMP which are not inside OPEN //
FimpNIOpen = Not Inside FIMP OPEN
//END auxiliary layers //
M181
{
    @N-WELL spacing to Open Area must be >= 35
    EXTERNAL NTUB OPEN < 35 SINGULAR
}
M182
{
    @Must be no coincidence between N-WELL and Open Area
    CUT NtubNIOpen OPEN
}
M221
{
    @Must be no coincidence between N-WELL IMPLANT and Open Area
    CUT FimpNIOpen OPEN
}
//////////      END N-WELL + N-WELL IMPLANT (NTUB)          //////////
/////////////////////////////////////////////////////////////////

/////////////////////////////////////////////////////////////////
//////////      DIFFUSION/ACTIVE AREA (DIFF)          //////////

//M203
//{
// @space between marging of 2 Diffusion must be >= 10
// EXTERNAL DIFF < 10 SINGULAR
//}

```

```

M204
{
  @Diffusion spacing to Open Area must be >= 35
    EXTERNAL DIFF OPEN < 35 SINGULAR
}
M208
{
  @Must be no coincidence between Diffusion and Open Area
    AND DIFF OPEN
}

////////// END DIFFUSION/ACTIVE AREA (DIFF)          //////////
////////////////////////////////////

////////////////////////////////////
////////// POLYSILICON 1 (POLY1)                      //////////

// auxiliary layers //

// The Layer Poly1NIOpen contains all the polygons of POLY1 which are not inside OPEN //
Poly1NIOpen = Not Inside POLY1 OPEN

//END auxiliary layers //
M242
{
  @Polysilicon 1 spacing to Open Area must be >= 3
    EXTERNAL POLY1 OPEN < 30 SINGULAR
}
M243
{
  @Must be no coincidence between Polysilicon 1 and Open Area.
    CUT Poly1NIOpen OPEN
}

////////// END POLYSILICON 1 (POLY1)          //////////
////////////////////////////////////

////////////////////////////////////
////////// POLYSILICON 2 (POLY2)                      //////////

// auxiliary layers //

// The Layer Poly2NIOpen contains all the polygons of POLY2 which are not inside OPEN //
Poly2NIOpen = Not Inside POLY2 OPEN

//END auxiliary layers //
M412
{
  @Polysilicon 2 spacing to Open Area must be >= 3 .
    EXTERNAL POLY2 OPEN < 3 SINGULAR
}
M413
{
  @Must be no coincidence between Polysilicon 2 and Open Area.
    CUT Poly2NIOpen OPEN
}
////////// END POLYSILICON 2 (POLY2)          //////////
////////////////////////////////////

////////////////////////////////////
////////// N + IMPLANT (NPLUS)                      //////////

// auxiliary layers //

// The Layer NplusNIOpen contains all the polygons of NPLUS which are not inside OPEN //
NplusNIOpen = Not Inside NPLUS OPEN

//END auxiliary layers //

```

```

M261
{
  @N + PLUS IMPLANT spacing to Open Area must be >= 35 .
  EXTERNAL NPLUS OPEN < 35 SINGULAR
}

M262
{
  @Must be no coincidence between N + IMPLANT and Open Area.
  CUT NplusNIOpen OPEN
}

//////////      END      N + IMPLANT (NPLUS)      //////////
//////////
//////////
//////////      P + IMPLANT (PPLUS)      //////////

// auxiliary layers //

// The Layer NplusNIOpen contains all the polygons of PPLUS which are not inside OPEN //
PplusNIOpen = Not Inside PPLUS OPEN

//END auxiliary layers //

M281
{
  @P + PLUS IMPLANT spacing to Open Area must be >= 35 .
  EXTERNAL PPLUS OPEN < 35 SINGULAR
}

M282
{
  @Must be no coincidence between P + IMPLANT and Open Area.
  AND PplusNIOpen OPEN
}

//////////      END      P + IMPLANT (PPLUS)      //////////
//////////
//////////
//////////      OPEN CONTACT/CONTACT (CONT)      //////////

M301
{
  @CONTACT spacing to Open Area must be >= 1.0 .
  EXTERNAL CONT OPEN < 1.0 SINGULAR
}

M309
{
  @Must be no coincidence between CONT and Open Area.
  AND CONT OPEN
}

//////////      END      OPEN CONTACT/CONTACT (CONT)      //////////
//////////
//////////
//////////      METAL 1 (MET1)      //////////

// auxiliary layers //

// The Layer Met1NIOpen contains all the polygons of METAL1 which are not inside OPEN //
Met1NIOpen = Not Inside MET1 OPEN

//END auxiliary layers //

M322
{
  @Must be no coincidence between Polysilicon 1 and Open Area.
  CUT Met1NIOpen OPEN
}

M323
{
  @METAL 1 spacing to Open Area must be >= 3.0 .
  EXTERNAL MET1 OPEN < 3.0 SINGULAR
}

```

```

//////////          END   METAL1 (MET1)          //////////
////////////////////////////////////

////////////////////////////////////
//////////          METAL 2 (MET2)          //////////

// auxiliary layers //

// The Layer Met2NIOpen contains all the polygons of METAL2 which are not inside OPEN //
Met2NIOpen = Not Inside MET2 OPEN

//END auxiliary layers //
M362
{
  @Must be no coincidence between METAL2 and Open Area.
    CUT Met2NIOpen OPEN
}

M363
{
  @METAL 2 spacing to Open Area must be >= 3.0 .
    EXTERNAL MET2 OPEN < 3.0 SINGULAR
}
//////////          END   METAL1 (MET1)          //////////
////////////////////////////////////

////////////////////////////////////
//////////          OPEN VIA (DVIA)          //////////
M3411
{
  @Must be no coincidence between VIA and Open Area.
    AND DVIA OPEN
}

M341
{
  @OPEN VIA spacing to Open Area must be >= 1.0 .
    EXTERNAL DVIA OPEN < 1.0 SINGULAR
}
//////////          END OPEN VIA (DVIA)          //////////
////////////////////////////////////

////////////////////////////////////
//////////          PASSIVATION OPENNING (PAD)          //////////
M386
{
  @Must be no coincidence between PASSIVATION and Open Area.
    AND PAD OPEN
}

M383
{
  @PASSIVATION spacing to Open Area must be >= 35 .
    EXTERNAL PAD OPEN < 35 SINGULAR
}
//////////          END PASSIVATION OPENNING (PAD)          //////////
////////////////////////////////////

```

Annexe D : Méthodes et algorithmes d'optimisation

1. Introduction

Un problème d'optimisation sous contrainte dans \mathfrak{R}^n est présenté de la forme :

$$(P) \begin{cases} \text{Minimiser } f(x) \\ \text{sous les Contraintes :} \\ g_i(x) \leq 0 \quad i = 1..m \\ x \in S \subset \mathfrak{R}^n \end{cases}$$

Si f est une fonction scalaire à variable scalaire et si le problème ne fait pas intervenir de contrainte, on parlera d'optimisation unidimensionnelle sans contrainte : l'étude des techniques de résolution pour ces problèmes fera l'objet de la section suivante. Dans la section 3, nous traiterons de l'optimisation d'une fonction à plusieurs variables sans contrainte. Enfin, le cas général sera abordé à la dernière section.

1.1. Convergence

Excepté pour certains cas élémentaires comme la minimisation d'une fonction quadratique à une variable, il est généralement impossible de résoudre un Programme Mathématique de façon analytique.

Pour obtenir la solution optimale x^* d'un tel problème, on utilise donc principalement des méthodes algorithmiques qui procèdent par construction d'une suite de valeurs (x_k) convergeant vers x^* . Ceci conduit naturellement à définir les notions de convergence et de vitesse de convergence comme critère d'évaluation d'un algorithme de Programmation Mathématique.

1.1.1. Définition

Une suite (x_k) de valeurs converge vers x^* si²⁰:

$$\forall \varepsilon, \exists N, k \geq N \Rightarrow \|x_k - x^*\| < \varepsilon$$

1.1.1. Définition

Soit (x_k) une suite de valeurs convergeant vers x^* et $p \in \mathbb{N}^*$.

Soit :

$$\beta = \lim_{k \rightarrow \infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|^p}$$

L'ordre de convergence de (x_k) est la valeur minimale de p telle que (x_k) est bornée.

Si p est égal à 1 alors β est le taux de convergence de la suite.

On définit parfois :

$$\beta = \lim_{k \rightarrow \infty} \frac{\|f(x_{k+1}) - f(x^*)\|}{\|f(x_k) - f(x^*)\|^p}$$

La plupart des algorithmes de Programmation Mathématique n'éprouvent pas de difficulté pour converger vers une valeur fixe x^* . Si cette valeur x^* approximée correspond généralement à la valeur

²⁰ $\|\cdot\|$ désigne la norme euclidienne dans \mathfrak{R}^n

d'un optimum *local*, il n'est pas toujours évident que la valeur obtenue constitue l'optimum *global* : le paragraphe suivant se propose d'illustrer cette difficulté supplémentaire.

1.2. Optimum local et optimum global

Pour simplifier, nous considérons désormais le Programme Mathématique :

$$\min_{x \in S} f(x)$$

1.1.2. Définition

On dit que la fonction f admet un minimum global x^* sur le domaine S si $f(x) \geq f(x^*)$ pour tout $x \in S$.

1.1.2. Définition

Supposons que f soit définie en tout point du voisinage de x^* .

La fonction f admet un *minimum local* x^* s'il existe $\varepsilon, 0 < \varepsilon < \delta$, tel que pour tout, $x \in S$

$$\|x - x^*\| < \varepsilon, f(x) \geq f(x^*).$$

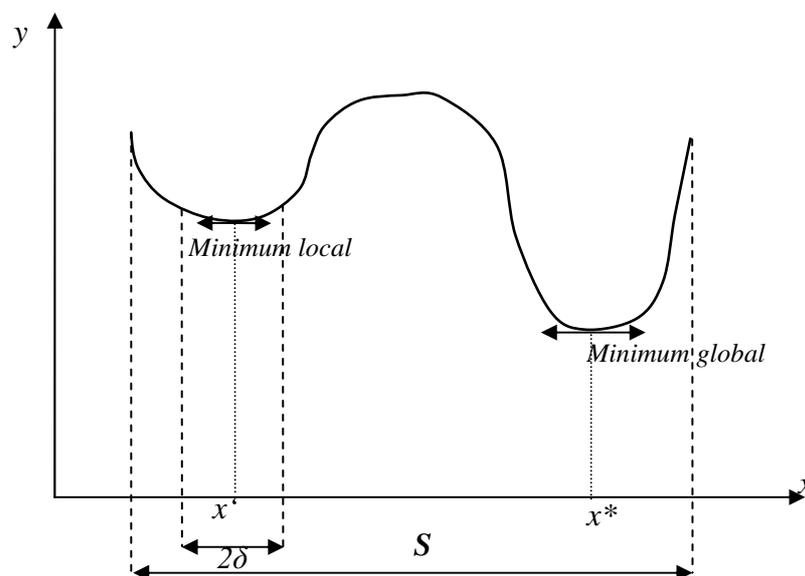


Figure.1. Minimum Local & Minimum Global

On remarque en particulier que le minimum global de la fonction figure nécessairement parmi les minima locaux. En général, il n'est pas difficile de caractériser un minimum local par des conditions nécessaires et/ou suffisantes. Il n'en est pas de même pour le minimum global, sauf lorsque la fonction et les contraintes vérifient des propriétés particulières (cas des Programmes Mathématiques *Convexes*).

1.3. Convexité

1.1.3. Définition

Un ensemble $S \subset \mathfrak{R}^n$ est dit convexe si et seulement si :

$$\forall x, y \in S, \forall \lambda \in [0, 1], \lambda \cdot x + (1 - \lambda) \cdot y \in S$$

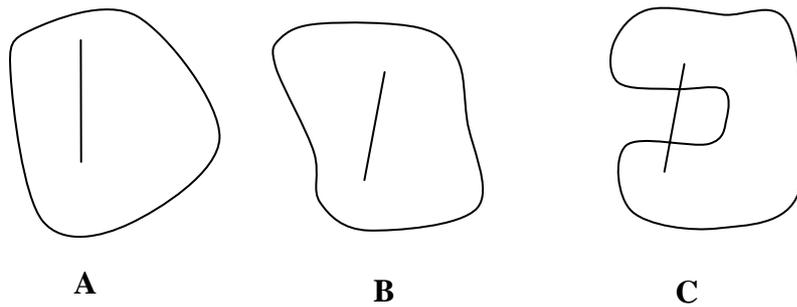


Figure.2. Exemples de domaines convexes & non-convexes.

1.1.3. Définition

Soit $S \subset \mathfrak{R}^n$ et $f : S \rightarrow \mathfrak{R}$

On dit que la fonction f définie sur S convexe est *convexe* si :

$$\forall x, y \in S, \forall \lambda \in [0,1], f(\lambda \cdot x + (1-\lambda) \cdot y) \leq \lambda \cdot f(x) + (1-\lambda) \cdot f(y)$$

La fonction f est concave si $-f$ est convexe.

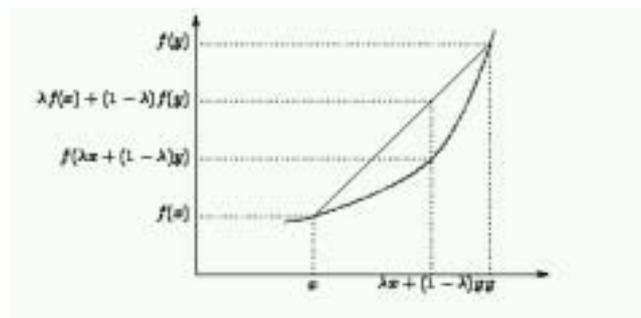


Figure.3. Fonction convexe

On définit alors la notion de Programme Convexe :

1.1.3. Définition

Un Programme Mathématique est dit *convexe* s'il consiste à *minimiser une fonction convexe* (resp. maximiser une fonction *concave*) sur un *domaine convexe*.

Par exemple, si la fonction f et les fonctions g_i sont convexes et si le domaine S est convexe, alors le Programme Mathématique (P) de la section 1.2 est un Programme Convexe.

Les propriétés particulières des fonctions convexes définies sur un domaine convexe permettent d'établir le théorème suivant :

Théorème

Pour un Programme Convexe, tout optimum local est un optimum global.

Par conséquent, pour un Programme Convexe, les conditions nécessaires et/ou suffisantes caractérisant un optimum local caractérisent également un optimum global.

2. Optimisation Unidimensionnelle

La première partie de cette section recense l'exposé des principales notions mathématiques qui permettent de caractériser un optimum local ou global d'une fonction à une variable. Les principales méthodes de résolution numériques sont présentées en seconde partie. Nous nous plaçons dans le cadre de la minimisation d'une fonction.

2.1. Outils Mathématiques

1.2.1. Dérivée et sous-gradient

1.2.1.1. Définition

Soit $f : S \subset \mathfrak{R} \rightarrow \mathfrak{R}, x_0 \in S$ et $\alpha = \lim_{\substack{x \rightarrow x_0 \\ x \in S \setminus \{x_0\}}} \frac{f(x) - f(x_0)}{x - x_0}$

On dit que la fonction f est dérivable au point x_0 lorsque α existe. Cette limite est appelée dérivée de f au point x_0 et se note $f'(x_0)$

Si on note Γ_f le graphe de la fonction f , on peut interpréter géométriquement la dérivée comme étant égale à la valeur de la pente de la tangente à Γ_f en x_0 . Au voisinage de x_0 la fonction f sera donc croissante si sa dérivée est positive, décroissante si elle négative et stationnaire sinon.

Si f admet une dérivée en tout point de S , on dira que f' est la fonction dérivée de f . Cette fonction peut elle-même admettre une dérivée qui sera notée f'' , etc.

Pour une fonction convexe dérivable en x_0 , on peut établir l'inégalité fondamentale :

$$f(x) \geq f(x_0) + f'(x_0) \cdot (x - x_0)$$

Lorsqu'une fonction f convexe n'est pas dérivable en un point x_0 , on substitue la notion de sous-gradient à celle de dérivée.

1.2.1.1. Définition.

On appelle sous-gradient de la fonction f au point x_0 tout scalaire $\gamma \in \mathfrak{R}$ vérifiant :

$$f(x) \geq f(x_0) + \gamma \cdot (x - x_0)$$

L'ensemble de tous les sous-gradients de f en x_0 sera appelé sous-différentiel de f au point x_0 et noté $\delta f(x_0)$.

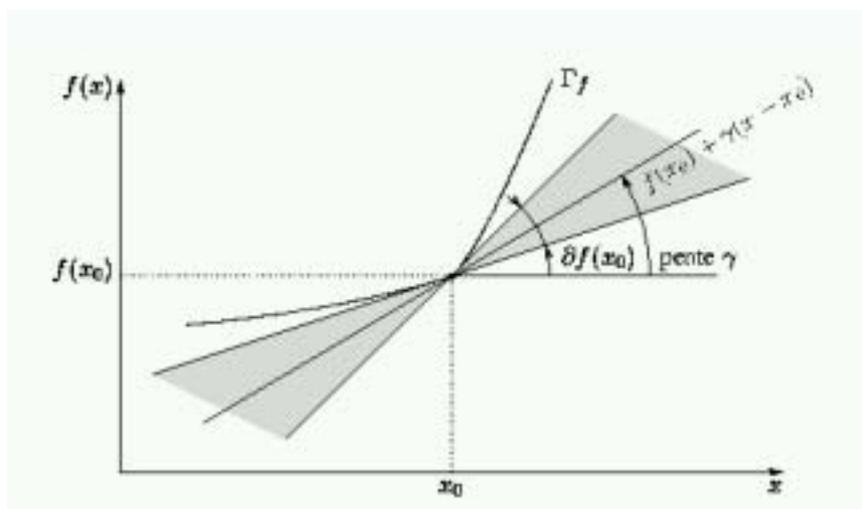


Figure.4. Sous-gradient, sous-différentiel.

Il existe une méthode simple pour vérifier si une fonction f est convexe lorsque celle-ci est deux fois dérivable :

Théorème

Soit $f : S \rightarrow \mathfrak{R}$ admettant une dérivée seconde pour tout $x \in S$.

La fonction f est convexe sur S si et seulement si, pour tout $x \in S$, $f''(x) \geq 0$.

2.2. Caractérisations de l'optimum

Dérivée et sous-gradient permettent d'établir quelques propriétés vérifiées par l'optimum lorsque f est dérivable ou convexe. La première d'entre-elles est une condition nécessaire d'optimalité lorsque f est dérivable :

Théorème

Soit f admettant une dérivée au point x_0 .

Pour que la fonction f admette un optimum local en x_0 , il faut : $f'(x) = 0$

Si la dérivée seconde est disponible, on peut sensiblement renforcer la condition d'optimalité :

Théorème.

Soit f admettant une dérivée seconde au point x_0 .

Pour que la fonction f admette un minimum local en x_0 , il faut : $f'(x) = 0$ et $f''(x) \geq 0$

En particulier, on peut établir la condition suffisante suivante :

Théorème.

Soit f admettant une fonction dérivée seconde définie sur S .

Si $f'(x) = 0$ et $f''(x) > 0$ alors x_0 est un minimum local pour f .

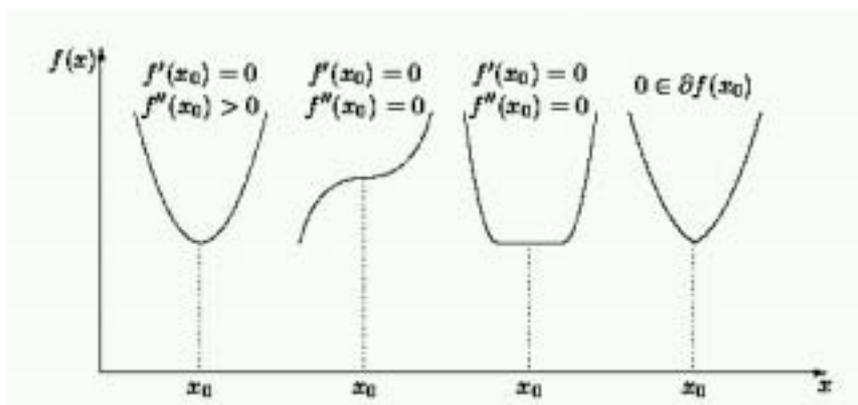


Figure.5. Dérivée, sous-gradient et minimum.

Enfin, comme indiqué précédemment, la convexité de la fonction garantit de « bonnes » propriétés de l'optimum :

Théorème

Soit f une fonction convexe sur S .

La fonction f admet un minimum global en x_0 si et seulement si $0 \in \delta f(x_0)$

2.3. Méthodes de Résolution Numériques

Les méthodes de résolution numériques se divisent en deux classes principales : les algorithmes utilisant la dérivée -- méthode de NEWTON-RAPHSON pour les fonctions admettant une dérivée seconde, méthode de la sécante pour les fonctions admettant uniquement une dérivée première -- et les techniques procédant uniquement par segmentation d'intervalle (sans utiliser la dérivée).

1.2.3. Dérivée connue et continue

1.2.3.1. Méthode de NEWTON-RAPHSON

Cette méthode s'applique aux fonctions dont on connaît la dérivée seconde en tout point. Rappelons que, lorsque f est dérivable, tout optimum local x^* vérifie $f'(x^*) = 0$

La méthode de NEWTON-RAPHSON consiste à construire une suite (x_k) de valeurs convergeant progressivement vers un point où la fonction f' dérivée s'annule (et donc vers un optimum local).

Soit x_k le point obtenu à l'étape k . La formule de TAYLOR-YOUNG permet d'approximer la fonction f' au voisinage de x_k par : $f'(x) = f'(x_k) + (x - x_k).f''(x_k) + o(x - x_k)$.

En d'autre termes, la fonction est f' approximée au voisinage x_k de par la tangente à la courbe en x_k d'équation $y = f'(x_k) + (x - x_k).f''(x_k)$

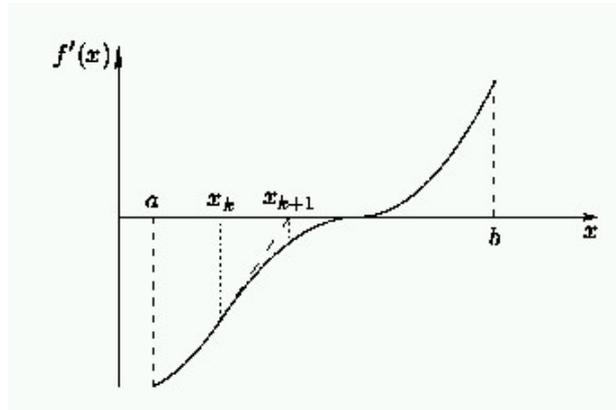


Figure.6. Méthode de NEWTON-RAPHSON.

Le point x_{k+1} est choisi comme l'intersection de cette droite avec l'axe (Ox) :

$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}$$

On peut effectuer les quelques remarques suivantes :

- si la fonction f est quadratique, la méthode converge en 1 itération ;
- la convergence globale n'est pas garantie ;
- localement, la convergence de la méthode est quadratique²¹.

1.2.3.1. Méthode de la sécante

Si on ne dispose pas de la dérivée seconde, ou si l'on souhaite économiser le coût de calcul de cette dérivée seconde, il est possible d'approximer la dérivée seconde en x_k par :

$$f''(x_k) \approx \frac{f'(x_k) - f'(x_{k-1})}{x_k - x_{k-1}}$$

Ainsi on peut citer:

$$x_{k+1} = x_k - f'(x_k) \cdot \frac{x_k - x_{k-1}}{f'(x_k) - f'(x_{k-1})}$$

Pour résoudre $f'(x) = 0$, on approxime donc la fonction $f'(x)$ non plus par sa tangente mais par la droite passant par les points $((x_{k-1}, f'(x_{k-1}))$ et $((x_k, f'(x_k)))$.

1.2.3. Dérivée inconnue

Dans ce paragraphe, nous décrivons une méthode plus générale que celles vues précédemment. En effet, elle ne nécessite pas d'hypothèse particulière de continuité ou de différentiabilité. Nous supposons simplement que la fonction f est une fonction *unimodale*.

1.2.3.2. Définition

On dit que la fonction f est *unimodale* sur l'intervalle réel $[a, b]$ si elle admet un minimum $x^* \in [a, b]$ si :

²¹ Au voisinage de l'optimum, on a : $|x_{k+1} - x^*| = C|x_k - x^*|^2$

$\forall x_1 \in [a, b], \forall x_2 \in [a, b],$ avec $x_1 < x_2$, on a :

- $x_2 \leq x^* \Rightarrow f(x_1) > f(x_2)$;
- $x_1 \geq x^* \Rightarrow f(x_1) < f(x_2)$.

La donnée de la valeur d'une fonction *unimodale* en deux points distincts de $[a; b]$ permet de localiser plus précisément la position du minimum, comme l'illustre la figure suivante :

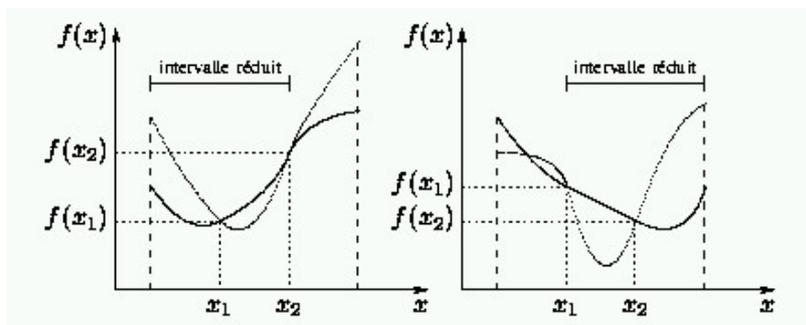


Figure.7. Fonction unimodale et minimum.

Les algorithmes de recherche de minimum s'appuient sur cette propriété pour réduire progressivement l'intervalle contenant le minimum, jusqu'à l'obtention d'un intervalle de longueur inférieure à la précision souhaitée.

Il est clair qu'il faut au minimum deux évaluations de la fonction pour pouvoir réduire l'intervalle initial. Par conséquent, la stratégie optimale (en terme de facteur de réduction de l'intervalle initial) consiste à placer les deux valeurs à proximité du centre de l'intervalle $[a; b]$. En itérant ce processus, on obtient la méthode dite de dichotomie.

1.2.3.2. Méthode de Dichotomie

Dans cette méthode, on commence par choisir une valeur ε la plus petite possible, et telle que l'on puisse distinguer les valeurs $f(x - \varepsilon)$ et $f(x + \varepsilon)$. L'algorithme va ensuite procéder par construction d'une suite d'intervalles $[a_k, b_k]$ de plus en plus réduits, jusqu'à obtenir un intervalle de longueur inférieure à la précision l souhaitée.

```

{ -- Initialisation -- }
Choisir  $\varepsilon > 0$  et  $l > 0$ 
 $k \leftarrow 1, [a_1, b_1] \leftarrow [a, b]$ 
{ -- Boucle Principale -- }
Tant que  $b_k - a_k \geq l$  faire
     $x_1 \leftarrow (b_k + a_k) / 2 - \varepsilon; x_2 \leftarrow (b_k + a_k) / 2 + \varepsilon$ 
    si  $f(x_1) < f(x_2)$  alors
         $a_{k+1} \leftarrow a_k; b_{k+1} \leftarrow x_2$ 
    sinon
         $a_{k+1} \leftarrow x_1; b_{k+1} \leftarrow b_k$ 
    fin si
     $k \leftarrow k + 1$ 
Fin tant que

```

Algorithme.1. Méthode de Dichotomie.

La méthode de Dichotomie est optimale vis à vis du nombre de réductions d'intervalles, mais nécessite deux calculs de fonction par itération.

La méthode permettant d'obtenir le meilleur coefficient de réduction de l'intervalle initial comparativement au nombre d'évaluations effectuées est la méthode de Fibonacci. Nous présentons ci-après une approximation de cette méthode : la méthode du Nombre d'Or.

1.2.3.2. Méthode du Nombre d'Or

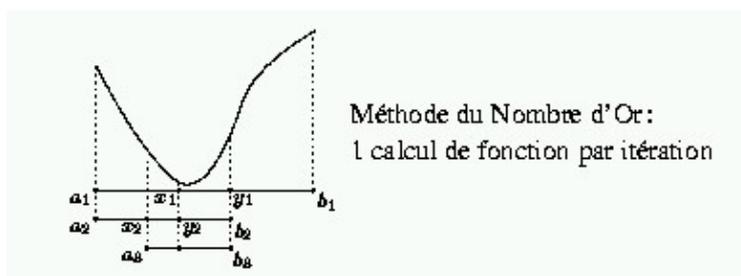


Figure.8. Méthode du Nombre d'Or.

```

{ -- Initialisation -- }
Choisir  $l > 0$ 
 $k \leftarrow 1, [a_1, b_1] \leftarrow [a, b], \alpha \leftarrow 0.618$ 
 $x_1 \leftarrow a_1 + (1 - \alpha)(b_1 - a_1), y_1 \leftarrow a_1 + \alpha(b_1 - a_1)$ 
{ -- Boucle Principale -- }
Tant que  $b_k - a_k \geq l$  faire
    Si  $f(x_k) < f(y_k)$  alors
         $a_{k+1} \leftarrow a_k; b_{k+1} \leftarrow y_k$ 
         $x_{k+1} \leftarrow a_{k+1} + (1 - \alpha)(b_{k+1} - a_{k+1}); y_{k+1} \leftarrow x_k$ 
    Sinon
         $a_{k+1} \leftarrow x_k; b_{k+1} \leftarrow b_k$ 
         $x_{k+1} \leftarrow y_k; y_{k+1} \leftarrow a_{k+1} + \alpha(b_{k+1} - a_{k+1})$ 
    Fin si
 $k \leftarrow k + 1$ 
Fin tant que

```

Algorithme.2. Méthode du Nombre d'Or.

3. Optimisation Non Linéaire Sans Contrainte

La plupart des concepts pour les fonctions à une variable peuvent s'étendre aux fonctions à plusieurs variables. Par exemple, la notion de gradient étend la notion de dérivée. L'articulation de ce chapitre est identique au précédent : après quelques préliminaires mathématiques, nous aborderons les techniques de résolution numériques proprement dites.

3.1. Outils Mathématiques

1.3.1. Gradient et sous-gradient

1.3.1.1. Définition.

Soit $f : S \subset \mathfrak{R}^n \rightarrow \mathfrak{R}$ S étant un ouvert dans \mathfrak{R}^n .

On dit que la fonction f admet en $x = (x_1, \dots, x_n)$ une i° dérivée partielle si la i° application partielle $(x_i \mapsto f(x))$ associée à f au point x est dérivable en x_i .

Cette i° dérivée partielle est notée $\frac{\delta}{\delta x_i} f(x)$

Si la fonction f admet n dérivées partielles en x , le vecteur $\left(\frac{\delta}{\delta x_1} f(x), \frac{\delta}{\delta x_2} f(x), \dots, \frac{\delta}{\delta x_n} f(x) \right)^T$ est appelé gradient de la fonction f au point x et noté $\nabla f(x)$.

Si la fonction f est convexe, on peut étendre l'inégalité du cas de fonction à une seule variable aux fonctions de plusieurs variables :

$$f(x) \geq f(x_0) + \nabla f^T(x_0) \cdot (x - x_0)$$

De la même façon, le sous-gradient d'une fonction de plusieurs variables est défini par :

1.3.1.1. Définition

On appelle sous-gradient de la fonction f au point x_0 tout vecteur $\gamma \in \mathfrak{R}^n$ vérifiant :

$$f(x) \geq f(x_0) + \gamma^T \cdot (x - x_0)$$

On dit que f admet des dérivées partielles secondes au point x si les n dérivées partielles $\frac{\delta}{\delta x_1} f, \frac{\delta}{\delta x_2} f, \dots, \frac{\delta}{\delta x_n} f$ admettent à leur tour des dérivées partielles en x .

Classiquement, on note: $\frac{\delta}{\delta x_i} \left(\frac{\delta}{\delta x_j} f \right) = \frac{\delta^2}{\delta x_i \delta x_j} f$ et $\frac{\delta}{\delta x_i} \left(-\frac{\delta}{\delta x_i} f \right) = \frac{\delta^2}{\delta x_i^2} f$

Ces dérivées partielles permettent de définir le Hessien de f au point x , noté $\nabla^2 f(x)$:

$$\nabla^2 f(x) = \begin{pmatrix} \frac{\delta^2}{\delta x_1^2} f(x) & \frac{\delta^2}{\delta x_1 \delta x_2} f(x) & \dots & \frac{\delta^2}{\delta x_1 \delta x_n} f(x) \\ \frac{\delta^2}{\delta x_2 \delta x_1} f(x) & \frac{\delta^2}{\delta x_2^2} f(x) & \dots & \frac{\delta^2}{\delta x_2 \delta x_n} f(x) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\delta^2}{\delta x_n \delta x_1} f(x) & \frac{\delta^2}{\delta x_n \delta x_2} f(x) & \dots & \frac{\delta^2}{\delta x_n^2} f(x) \end{pmatrix}$$

On peut remarquer que le Hessien est une matrice symétrique (car $\frac{\delta^2}{\delta x_i \delta x_j} f = \frac{\delta^2}{\delta x_j \delta x_i} f$).

Lorsque le Hessien est défini, on peut déterminer la convexité de la fonction f au travers du théorème :

Théorème

Soit $f : S \subset \mathfrak{R}^n \rightarrow \mathfrak{R}$ admettant un Hessien pour tout $x \in S$.

La fonction f est convexe sur S si et seulement si, pour tout $x \in S$, le Hessien $\nabla^2 f(x)$ est une matrice semi-définie positive ($\forall y, y^T \cdot \nabla^2 f(x) \cdot y \geq 0$).

1.3.1. Caractérisations de l'Optimum

Les théorèmes du cas de fonction à une seule variable se généralisent sans difficulté majeure :

Théorème (Condition Nécessaire)

Soit f admettant un vecteur gradient au point x_0

Pour que la fonction f admette un optimum local en x_0 , il faut : $\nabla f(x_0) = 0$

Théorème (Condition Nécessaire).

Soit f une fonction continue admettant des dérivées partielles premières et secondes continues pour tout $x \in \mathfrak{R}^n$.

Pour que la fonction f admette un minimum local en x_0 , il faut :

- $\nabla f(x_0) = 0$

- le Hessien $\nabla^2 f(x_0)$ est une matrice semi-définie positive.

Théorème (Condition Suffisante)

Soit f une fonction vérifiant les conditions du théorème précédente. Une Condition Suffisante pour que x_0 soit un minimum local est que :

$$- \nabla f(x_0) = 0$$

- le Hessian $\nabla^2 f(x_0)$ est une matrice définie positive ($\forall y \neq 0, y^T \cdot \nabla^2 f(x) \cdot y > 0$).

Le théorème dans le cas de fonction à une seule variable reste inchangé : si f est une fonction convexe, alors cette fonction admet un minimum global en x_0 si et seulement si 0 est un sous-gradient admissible pour f en ce point.

3.2. Méthodes de Résolution Numériques

1.3.2. Classification des méthodes d'optimisation

En prenant comme critères de classification la nature des variables (entière, réelle) et le type de contraintes d'un problème d'optimisation on peut dessiner l'arbre suivant dans une tentative de représenter l'ensemble des approches possibles :

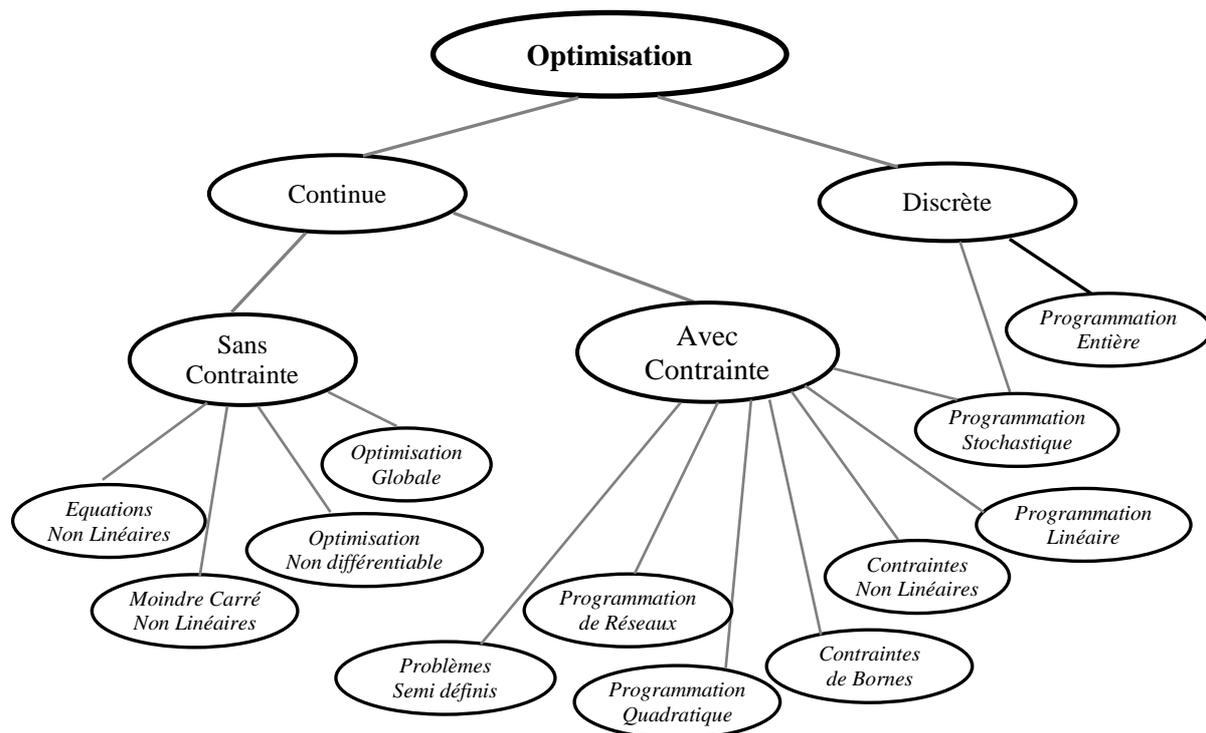


Figure.9. Arbre d'optimisation [1].

Discrète : Les variables sont à valeurs entières (\mathbb{Z}) :

Programmation entière (Integer Programming).

Programmation stochastique (Stochastic Programming).

Continue : Les variables sont à valeurs réelles (\mathbb{R}).

Sans contraintes :

Equations non linéaires.

Moindre carré non linéaire.

Optimisation Global.

Optimisation non différentiable.

Avec Contraintes :

Programmation linéaire.

Problèmes semi-définis.

Contraintes non linéaires.

Contraintes de bornes.

Programmation quadratique.

Programmation de réseaux.

Programmation Stochastique.

Il n'est pas possible de discuter en détail toutes ces familles de méthodes d'optimisation. Détaillons les approches qui nous semblent les plus intéressantes dans le domaine des MEMS.

1.3.2. Méthodes discrètes

On ne rencontre pratiquement jamais un problème à variables purement discrètes dans l'optimisation des microsystèmes. Il s'agit très souvent d'un problème mixte contenant à la fois des variables continues et d'autres discrètes. Le traitement de ces problèmes ne peut se faire que dans le cadre d'une optimisation discrète car ces méthodes permettent la prise en compte des variables continues en modifiant légèrement l'algorithmique sous-jacente. L'inverse, qui consiste à faire supporter à un algorithme d'optimisation à variables continues les variations discrètes de certaines variables, est autrement plus délicat, voire infaisable.

L'exemple type de l'utilisation de ces méthodes est lors de la conception des résonateurs de type comb-drive où le nombre des franges inter-digité est toujours entier positif tandis que les dimensions géométriques sont à valeurs réelles positives.

1.3.2.2. Programmation linéaire

La formulation mathématique de ce genre de problèmes peut s'écrire sous la forme :

$$\text{Min } \{c^T x ; Ax = b, x \geq 0, x \in Z^n\}$$

où Z^n est l'ensemble des vecteurs d'entiers relatifs de dimension n , A est une matrice réelle $p \times n$ et b est un vecteur réel à p dimension. A noter que la formulation indiquée précédemment est celle dite standard et on démontre que les autres problèmes apparemment différents (ne contenant pas par exemple $x \geq 0$) peuvent se ramener à cette écriture moyennant une transformation sur les variables.

Un algorithme typique pour résoudre cette problématique est celui dit du simplexe.

1.3.2.2. Programmation stochastique (valable pour discrète et continue)

Ces techniques sont utilisées dans les situations où l'on n'a pas une connaissance précise des données du problème à résoudre. C'est typiquement le cas dans une fonderie où une conception n'est jamais fabriquée exactement avec les dimensions et propriétés demandées par le concepteur. Les indications fournies par les technologues concernant par exemple le module de Young d'un matériau sont un ensemble de mesures que l'on représente sous forme de distribution statistique avec une moyenne et un écart-type.

L'idée fondamentale derrière ces techniques est de définir un recours. Le recours dans ce contexte est l'habilité à entreprendre une action corrective suite à l'avènement d'un événement aléatoire. Un exemple d'un recours à deux étapes est le suivant :

- choisir un ensemble de variables, X , pour contrôler ce qui se passe aujourd'hui.
- durant la nuit, un événement aléatoire arrive.
- demain, exécuter un recours, Y , pour corriger les éventuels effet néfaste de l'événement aléatoire.

Il est à noter que dans le cas des problèmes stochastiques linéaires, une formulation déterministe est possible. Elle revient à construire un nombre fini de scénarios et à programmer une solution optimale pour chaque d'entre eux.

1.3.2. Méthodes pour variables continues

Cette catégorie de problème est essentielle dans le développement des codes d'optimisation car beaucoup de codes d'optimisation à contrainte sont des extensions de logiciels d'optimisation non contrainte. Il n'y a guère que les algorithmes de moindre carré non linéaire et d'équations non linéaires qui présentent une rupture avec l'optimisation non contrainte.

Une optimisation sans contrainte consiste à chercher un minimum local d'une réelle fonction $f(x)$ où x est un vecteur de réels de dimension n . En d'autres termes, on recherche un vecteur x^* tel que $f(x^*) < f(x)$ pour tous les x « proches » de x^* .

L'optimisation globale qui consiste à trouver x^* tel que $f(x^*) < f(x)$ pour toutes les valeurs de x est un problème autrement plus difficile et il n'y a pas à ce jour une algorithmique efficace pour résoudre ce problème. Néanmoins, trouver un minimum local est suffisant dans la plupart des applications d'intérêt pratique.

Parmi les méthodes disponibles nous pouvons faire une distinction entre les méthodes de Newton et les méthodes dites Quasi-Newton.

Les méthodes de Newton sont des algorithmes nécessitant le calcul du vecteur gradient (vecteur de dérivées premières) et de la matrice hessienne (matrice dérivées seconde) de la fonction f à minimiser. Ces méthodes construisent une approximation (modèle) quadratique de la fonction objective autour de l'itéré x_k du vecteur des variables. Ce modèle est donné par :

$$q_k(s) = f(x_k) + \Delta(f) \cdot s + 0.5s^T \cdot B(x_k) \cdot s$$

L'opérateur est le produit scalaire et B est la matrice hessienne.

Différentes méthodes de newton ont été développé. La différence entre elles réside uniquement dans le calcul d'un s_k minimisant $q_k(s)$. Dans sa variante la plus simple où l'on suppose B localement définie positive s_k peut être la solution du système linéaire $n \times n$ suivant :

$$B(x_k) \cdot s_k = -\Delta f(x_k).$$

et l'itéré suivant sera alors

$$x_{k+1} = x_k + s_k$$

Il faut néanmoins noter que, même si la théorie prédit une convergence quadratique, la méthode basique de Newton a de sérieux problèmes de convergence [36].

Deux familles de techniques ont été développé pour améliorer la convergence : la recherche linéaire "*line-search*" et les régions de confiance.

Les recherches linéaires ne prennent pas le gradient comme direction de descente mais modifie cette direction pour trouver une direction de descente plus optimale. Elles essaient ensuite plusieurs pas dans cette direction de façon à garantir non seulement la descente de f mais aussi que cette descente soit au moins sur une fraction significative de la direction trouvée.

Les méthodes de région de confiance utilise le modèle quadratique original mais obligent x à rester dans un voisinage de x_k . Pour trouver le pas à l'itéré suivant, elle minimise la forme quadratique uniquement sur un certain voisinage défini par le rayon de confiance. Ce rayon de confiance est adapté automatiquement à chaque itération.

Dans le problème où la matrice hessienne n'est pas accessible ou trop coûteuse à calculer, la matrice exacte est remplacée par une approximation raisonnable. Deux sortes d'approches sont utilisées dans ce cas :

- Utiliser une différence de dérivées premières pour estimer les éléments de B. On exploite le fait que chaque colonne de B peut être approchée en prenant la différence entre deux instances des vecteurs gradient calculées dans deux points voisins. Pour les matrices creuses, les colonnes de la hessienne sont approchées par une seule évaluation du gradient en un point judicieusement choisi.
- Recourir aux méthodes dites Quasi-Newton où l'approximation de la matrice B est calculée en regardant une trace de la différence de gradients entre deux itérations consécutives (Davidon-Fletcher-Powell : DFP, Broyden-Fletcher-Goldfarb-Shanno : BFGS).

Signalons que des approches autres que celles dites de newton ont été développées. Nous pouvons en citer deux :

- Gradient conjugué non linéaire qui utilise des directions de recherche combinant la direction du gradient négatif à une autre choisie de telle sorte que la recherche linéaire se fasse sur une direction non précédemment explorée par l'algorithme. Cette méthode offre l'avantage d'exiger uniquement l'évaluation du gradient et d'être économique en terme de stockage mémoire.
- Méthode du simplex non linéaire où ni le gradient ni la hessienne ne sont requis. Cette méthode peut être utile pour minimiser les fonctions qui ne sont pas suffisamment régulière (cas peu fréquent pour le MEMS mais qui peut se présenter).

1.3.2.3. Fonction différentiable

Dans cette section, nous considérons que la fonction f admet un Hessien défini positif. Deux principales classes de méthode sont utilisées : les méthodes de descente et les méthodes dites du second ordre.

Méthodes de descente

Les méthodes de descente construisent la suite (x_k) à l'aide du gradient ∇f . Au point x_k une direction améliorante est en effet définie par $d_k = -\nabla f(x_k)$: le point x_{k+1} est alors obtenu en effectuant un déplacement suivant cette direction. Un déplacement est a priori intéressant s'il préserve :

$$f(x_k) \geq f(x_{k+1})$$

La plus élémentaire des méthodes de descente est la méthode de gradient.

```
{ -- Initialisation -- }
Choisir un point de départ  $x_0$ ,  $k \leftarrow 0$ ,
{ -- Boucle Principale -- }
Tant que Test d'arrêt non vérifié faire
     $d_k \leftarrow -\nabla f(x_k)$ 
     $x_{k+1} \leftarrow x_k + \lambda_k \cdot d_k / \|\nabla f(x_k)\|$ 
     $k \leftarrow k+1$ 
Fin tant que
```

Algorithme.3. Méthode de gradient.

La suite λ_k est en général choisie comme suit :

$$\begin{cases} \lambda_k \rightarrow 0 & (k \rightarrow \infty) \\ \sum_{k=0}^{\infty} \lambda_k = +\infty \end{cases}$$

La méthode de la plus forte pente est une première amélioration de la technique de base. On choisit λ_k de façon à minimiser la fonction de λ : $g(\lambda) = f[x_k - \lambda \nabla f(x_k)]$. Ceci permet d'obtenir un point x_{k+1} « optimal » vis-à-vis de la direction (unidimensionnelle) considérée.

Pour éviter un comportement en zig-zag de la méthode précédente, il est possible de générer à chaque étape k , p nouveaux points, et de prendre comme direction $x_{k+p} - x_k$: on parle alors de *méthode de plus forte pente accélérée*.

```
{ -- Initialisation -- }
Choisir un point de départ  $x_0$ ,  $k \leftarrow 0$ ,
{ -- Boucle Principale -- }
Tant que  $\|\nabla f(x_k)\| \geq \varepsilon$  faire
     $d_k \leftarrow -\nabla f(x_k)$ 
    Rechercher  $\lambda_k$  tel que :  $f(x_k + \lambda_k d_k) = \min_{\lambda \geq 0} f(x_k + \lambda d_k)$ 
     $x_{k+1} \leftarrow x_k + \lambda_k d_k$ 
     $k \leftarrow k+1$ 
Fin tant que
```

Algorithme.4. Méthode de la plus forte pente descendante.

Méthodes du second ordre

Sous certaines conditions (cf. formule de TAYLOR-YOUNG), on peut considérer qu'une fonction quelconque est approximée au voisinage d'un point par une forme quadratique. Par conséquent, on peut supposer qu'une méthode garantissant un bon comportement pour une fonction quadratique permettra d'obtenir de bonnes performances pour une fonction suffisamment régulière (au moins au

voisinage de l'optimum). Les méthodes spécifiques basées sur ces considérations sont regroupées sous le terme générique de méthodes du second ordre.

Considérons la forme quadratique :

$$q(x) = \frac{1}{2} x^T A x + b^T x + c$$

On souhaite déterminer le minimum global de cette fonction (la matrice symétrique $A = \nabla^2 q(x)$ étant par hypothèse supposée définie positive).

Les méthodes du second ordre permettent d'obtenir le minimum d'une telle fonction en n itérations (n désigne la dimension de l'espace de départ) à l'aide du concept de directions conjuguées. Deux directions d_1 et d_2 sont conjuguées, relativement à la forme q , si elles vérifient : $d_1^T A d_2 = 0$.

Plus généralement, k vecteurs d_1, \dots, d_k seront dits conjugués s'ils sont linéairement indépendants et si $d_i^T A d_j = 0$ pour tout $i \neq j$.

On peut alors établir le théorème clé suivant :

Théorème

Soient d_0, \dots, d_{n-1} , n directions conjuguées et x_0 un point arbitraire.

Pour tout $k \in [0; n[$, λ_k est la solution optimale du problème $\min_{\lambda \in \mathbb{R}} q(x_k + \lambda d_k)$ et x_{k+1} vaut $x_k + \lambda_k \cdot d_k$

On a alors :

$$-\nabla q(x_{k+1})^T \cdot d_j = 0, \text{ pour } j \in [0, k];$$

$$-\nabla q(x_0)^T \cdot d_k = \nabla q(x_k)^T \cdot d_k;$$

- x_{k+1} est la solution optimale du problème de minimisation de $q(x)$ relativement à $x - x_0 \in L(d_0, \dots, d_k)$, où $L(d_0, \dots, d_k)$ désigne le sous-espace engendré par les directions d_0, \dots, d_k .

En particulier, x_n est le minimum de q .

Les nombreuses méthodes basées sur ce résultat se distinguent au travers de la génération des directions conjuguées.

Dans la méthode de FLETCHER-REEVES, la construction de la base conjuguée s'effectue en utilisant la seule connaissance des gradients. La convergence s'effectue en n itérations pour une fonction quadratique.

```

{ -- Initialisation -- }
Choisir un point de départ  $x_0$ 
 $d_0 \leftarrow -\nabla f(x_0)$ 
 $k \leftarrow 0$ ,
{ -- Boucle Principale -- }
Tant que Test d'arrêt non vérifié faire
    Rechercher  $\lambda_k$  tel que :  $f(x_k + \lambda_k d_k) = \min_{\lambda \geq 0} f(x_k + \lambda d_k)$ 
     $x_{k+1} \leftarrow x_k + \lambda_k d_k$ 
     $\beta_k \leftarrow \frac{\|\nabla f(x_{k+1})\|^2}{\|\nabla f(x_k)\|^2}$ 
     $d_{k+1} \leftarrow -\nabla f(x_{k+1}) + \beta_k d_k$ 
     $k \leftarrow k + 1$ 
Fin tant que

```

Algorithme.5. Méthode de FLETCHER-REEVES.

Pour une forme quadratique définie positive, la connaissance du Hessian permet d'exprimer directement :

$$x^* = x - [\nabla^2 q(x)]^{-1} \cdot \nabla q(x)$$

La méthode de NEWTON utilise directement ce résultat en effectuant :

$$x_{k+1} = x_k - [\nabla^2 q(x_k)]^{-1} \cdot \nabla q(x_k)$$

Cette méthode converge clairement en une itération pour une fonction quadratique, mais présente l'inconvénient de faire appel au Hessien. La méthode de Broyden-Fletcher-Goldfarb-Shanno : BFGS permet de construire $[\nabla^2 q(x_k)]^{-1}$ en faisant uniquement appel aux gradients.

```

{ -- Initialisation -- }
Choisir un point de départ  $x_0$  et  $H_0$  définie positive
 $k \leftarrow 0$ ,
{ -- Boucle Principale -- }
Tant que Test d'arrêt non vérifié faire
     $d_k \leftarrow -H_k \cdot \nabla f(x_k)$ 
    Rechercher  $\lambda_k$  tel que :  $f(x_k + \lambda_k d_k) = \min_{\lambda \geq 0} f(x_k + \lambda d_k)$ 
     $x_{k+1} \leftarrow x_k + \lambda_k H_k \nabla f(x_k)$ 
     $\delta_k \leftarrow x_{k+1} - x_k, \gamma_k \leftarrow \nabla f(x_{k+1}) - \nabla f(x_k)$ 
     $\Delta_k \leftarrow (\delta_k \cdot \delta_k^T) / (\delta_k^T \cdot \gamma_k) - (H_k \cdot \gamma_k \cdot \gamma_k^T \cdot H_k) / (\gamma_k^T \cdot H_k \cdot \gamma_k)$ 
     $H_{k+1} \leftarrow H_k + \Delta_k$ 
     $k \leftarrow k + 1$ 
Fin tant que

```

Algorithme.6. Méthode de Broyden-Fletcher-Goldfarb-Shanno : BFGS.

Le terme Δ_k utilisé dans l'algorithme de Broyden-Fletcher-Goldfarb-Shanno : BFGS est appelé correction de rang 2. On peut montrer que les directions générées par l'algorithme sont conjuguées. En outre, pour une forme quadratique, on peut établir : $H_n = [\nabla^2 q(x)]^{-1}$.

1.3.2.3. Fonction quelconque

Lorsque les gradients sont inconnus, on doit se contenter de méthodes dites heuristiques. Les deux principales techniques sont celles de HOOKE-JEEVES et de ROSENBROCK.

Méthode de HOOKE-JEEVES

La méthode de HOOKE-JEEVES est composée de deux étapes enchaînées :

- une étape locale ;
- une étape globale.

L'étape locale procède à une recherche variable par variable en procédant à des « sauts » le long des axes de l'espace de façon à déterminer un point améliorant dans le voisinage. Lorsque aucun point améliorant n'est trouvé (échec), l'incrément des sauts est réduit.

L'étape globale consiste à établir un nouveau point d'entrée de la procédure locale par interpolation linéaire des points obtenus à la fin des itérations locales k et $k-1$. On arrête le processus par un test sur la validité de la progression globale de la méthode.

Références

- [1] J.F Bonnans, J.C. Gilbert, C. Lemarechal, C. Sagastizabal, "Optimisation numérique aspects théoriques et pratiques", Springer, ISBN 3-540-63183-6, 1997.
- [2] L. Polak, "Optimization Algorithms and Consistent Approximations", Springer, ISBN 0-387-94971-2, 1997
- [3] P. E. Gill, W. Murray and M.H. Wright, "Practical Optimization", ACADEMIC PRESS ISBN 0-12-283952-8, 1999 (second edition).