

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
MINISTRE DE L'ENSEIGNEMENT SUPERIEUR ET DE  
LA RECHERCHE SCIENTIFIQUE  
UNIVERSITE M'HAMED BOUGARA BOUMERDES  
Faculté des Sciences de l'Ingénieur



Département Génie Mécanique  
Laboratoire d'Energétique, Mécanique et Ingénierie  
Groupe Modélisation En Mécanique et Productique

# MEMOIRE

Présenté par

**M. MANSOURI Khaled**

En vue de l'obtention du diplôme de

**MAGISTER en GENIE MECANIQUE**

OPTION : MODELISATION ET SIMULATION EN MECANIQUE

Modélisation et Optimisation Multi-Objectif  
d'une Structure Mécanique Poly-Articulée dédiée  
aux Machines à Commande Numérique Multi-axiales

SOUTENU LE 11/07/2010 DEVANT LE JURY SUIVANT :

Pr. Abdelatif ZERIZER	Professeur, UMB Boumerdès	Président
Pr. Kacem MANSOURI	Professeur, UMB Boumerdès	Examineur
Dr. Moussa HADDAD	MC(A), EM Polytechnique	Examineur
Dr. Kamel MOHAMMEDI	MC(A), UMB Boumerdès	Examineur
Dr. Idir BELAIDI	MC(A), UMB Boumerdès	Rapporteur

Année universitaire 2009/2010

## Résumé :

L'usinage à grande vitesse permet d'obtenir des pièces de meilleure qualité, de réduire les temps d'usinage et ainsi d'augmenter les cadences de production des machines. Les chercheurs ont puisé du domaine de la robotique une technologie pour trouver une machine outil plus performante. Le travail que nous avons fait est basé sur les structures parallèles pour l'usinage. Ces structures sont nouvelles pour les fabrications de machines-outils. Nous avons proposé dans notre travail un robot parallèle « Delta » et nous avons modélisé un problème d'optimisation multi-objectif, pour trouver une machine performant en fonction des exigences inhérentes à l'isotropie et à la dynamique. Dans ce travail La résolution du problème d'optimisation multi-objectif est basée sur la mise en œuvre de l'algorithme génétique NSGAI, implémenté sous Matlab.

**Mots clefs :** Système Poly-articulés, Machines à Commande Numériques, Modélisation Géométrique Et Cinématique, Optimisation Multi-Objectif.

## Abstract :

The high speed machining can get better quality parts, reduce machining time and thus increase production rates of machines. Researchers drew the field of robotics technology to find a machine more powerful tool. The work we have done is based on parallel structures for machining. These structures are novel for the manufacture of machine tools. We proposed in our work a parallel robot "Delta" and we have modeled a problem of multi-objective optimization to find a machine performance based on inherent requirements of isotropy and dynamics. In this work solving the problem of multi-objective optimization is based on the implementation of the genetic algorithm NSGA-II, implemented in Matlab.

**Key words:** System Poly-articulated, Machines With Numerical Control, Geometrical And Kinematic Modeling, Multi-Objective Optimization.

## ملخص:

عملية القطع ذات السرعة العالية تسمح بالحصول على قطع الغيار بجودة عالية، مع تقليل الوقت المحدد للتشكيل وبالتالي زيادة إنتاجية الآلات. الباحثون استمدوا من مجال الروبوتات تقنية للعثور على آلة ذات أداء أفضل. العمل الذي قمنا به مؤسس على هياكل موازية لآلات القطع. هذه الهياكل تعتبر جديدة لصناعة الآلات المكنية. اقترحنا في عملنا روبوت صاحب الهيكل الموازي "دلتا Delta" وعلى غرار ذلك لدينا مشكلة التحسين متعددة الأهداف للحصول على جهاز أكثر فعالية على أساس المتطلبات اللازمة كالتماثل والديناميك. في هذا العمل قمنا بحل مشكلة التحسين متعددة الأهداف باستخدام الخوارزمية الجينية NSGAI، المنجز في برنامج ماتلاب.

الكلمات المفتاحية : نظام متعدد المفاصل، آلات ذات تحكم رقمي، النمذجة الهندسية والحركية، التحسين متعدد الأهداف.

# *Remerciements*

Je remercie Dieu tout puissant clément et miséricordieux de m'avoir soigné et aidé.

Je tiens, avant tout, à exprimer ma profonde gratitude à monsieur **Dr. Idir BELAIDI**, maître de conférences à l'Université de M'hamed Bougara Boumerdes, qui a assumé la direction de ce travail. Qu'il veuille bien trouver ici l'expression de ma reconnaissance pour son dévouement, sa patience, sa disponibilité, ses conseils et son aide constant qu'il m'a apporté tout au long de ce travail.

Je tiens à remercier tout particulièrement le **Dr. Kamal MOHAMMEDI**, maître de conférences à l'Université de M'hamed Bougara Boumerdes, pour ses nombreux conseils et pour l'acuité scientifique qu'il a mise en œuvre lors de nos discussions.

Je remercie vivement le **Pr. Abdelatif ZERIZER** (UMBB) pour avoir bien accepté de présider ce jury.

Je remercie vivement les membres du jury **Pr. Kacem MANSOURI, Dr. Moussa HADDAD et Dr. Kamel MOHAMMEDI** pour avoir bien voulu accepter de faire partie du jury de soutenance. Qu'ils trouvent ici l'expression de ma gratitude.

J'adresse mes vifs remerciements à tous les enseignants qui par leur enseignement, leur encouragement et leur aide ont contribué à ma formation durant toutes mes études à l'Université de M'hamed Bougara Boumerdes.

Je tiens également à remercier tous mes collègues d'études en post graduation pour leurs soutiens et leur motivation.

Que tous mes amis sans exception, ils sont nombreux, je ne peux tous les citer mais je ne les oublie.

Enfin, que toutes les personnes ayant contribué, de près ou de loin, à la réalisation de ce travail, soient chaleureusement remerciées.

# *Dédicace*

Je dédie ce travail à ma mère pour ses encouragements et ses prières tout au long de mes études. A l'âme de mon père qui nous a planté l'amour de science.

Je le dédie à mes frères et sœurs, et je les remercie pour leurs encouragements et leurs aides ainsi que toute ma grande famille.

A tous mes amis sans citer les noms.

# Table des matières

---

Résumé	
Remerciements	
Dédicace	
Sommaire.....	i
Liste des figures.....	iv
Liste des tableaux.....	vi
Notations et Abréviations.....	vii
<b>Introduction générale</b> .....	<b>1</b>
<b>Chapitre I : Robots parallèles pour Machines outils</b>	
I.1. Introduction.....	3
I.2. Définitions.....	3
I.3. Présentation de quelques robots parallèles.....	4
I.4. Machines-outils parallèles.....	8
I.4.1. La machine Ingersoll H0H600.....	9
I.4.2. La machine Mikromat 6X Hexapode.....	9
I.4.3. La machine Cosmo Center PM-600 de la société Okuma.....	10
I.4.4. La machine HEXACT développée par INA et l'IFW.....	10
I.4.5. L'Hexapode 300 de la société CMW.....	11
I.4.6. La machine Tornado 2000 de la société Hexel.....	11
I.4.7. La machine HexaM de la société Toyoda.....	12
I.4.8. La machine Hexaglide de l'ETHH (Zurich).....	12
I.4.9. La machine Triaglide.....	13
I.4.10. La machine UraneSX de la société Renault Automation/Comau.....	13
I.4.11. La machine Dyna-M du WZL (Aix-la-Chapelle).....	14
I.5. La problématique de mécanisme parallèle.....	14
I.5.1. Introduction.....	14
I.5.2. Les différentes couches d'un système de robot parallèle.....	15
I.5.3. Le Mécanisme.....	16
I.5.4. Cinématique.....	17
I.5.5. Singularités.....	17
I.5.6. Espace de travail.....	18
I.5.7. Organisation du mouvement.....	19
I.5.8. Étalonnage.....	20
I.5.9. Dynamique.....	20
I.5.10. Synthèse.....	21
I.5.11. Contrôleur.....	21
I.6. Conclusion.....	22
<b>Chapitre II : Méthodes d'Optimisation multi-objectifs</b>	
II.1. introduction.....	23
II.2.1. Problème d'optimisation multi-objectif.....	23
II.2.2. Solution d'un problème d'optimisation multi-objectif.....	23
II.3.1. Les méthodes agrégées.....	26
II.3.1.1. La moyenne pondérée.....	26
II.3.1.2. Méthode du but programmé.....	26
II.3.1.3. Méthode du but à atteindre.....	26
II.3.1.4. Le min-max.....	27
II.3.1.5. La méthode $\varepsilon$ -contrainte.....	27
II.3.2. Les méthodes Pareto.....	27
II.3.3. les méthodes non agrégées, non Pareto.....	27

II.3.3.1.	Sélection parallèle (VEGA)	27
II.3.3.2.	Utilisation des genres	28
II.3.3.3.	La méthode lexicographique	28
II.4.1.	Principes généraux	29
II.4.2.	L'état de l'art de l'algorithme génétique multi-objectif	31
II.4.3.	Description détaillée	32
II.4.3.1.	Codage des données	32
II.4.3.2.	Génération aléatoire de la population initiale	33
II.4.3.3.	Gestion des contraintes	33
II.4.3.4.	Opérateur de croisement	34
II.4.3.5.	Opérateur de mutation	35
II.4.3.6.	Principes de sélection	36
II.5.	Présentation de l'algorithme NSGA II	36
II.5.1.	Non dominated Sorting Genetic Algorithm (NSGA)	37
II.5.2.	NSGA II	38
II.6.	Conclusion	42

### Chapitre III : Modélisation du robot Delta

III.1.	introduction	43
III.2.	Architectures parallèles candidates pour les machines-outils	43
III.2.1.	Limitations de machines outil d'une série	43
III.2.2.	Choix d'une architecture	43
III.3.	Modélisation générique	45
III.3.1.	Paramétrage	45
III.3.2.	la description géométrique	45
III.3.3.	Modélisation géométrique	47
III.3.3.1.	Modèle géométrique inverse	47
III.3.3.2.	Modèle géométrique direct	48
III.3.4.	Modélisation cinématique	48
III.3.5.	Choix d'un agencement	49
III.3.6.	Réduction du nombre de paramètres	50
III.4.	Modélisation de l'architecture choisie	51
III.4.1.	Paramétrage	51
III.4.2.	Modèles géométriques	51
III.4.3.	Modèles cinématiques	53
III.4.4.	Modèles dynamiques	54
III.5.	Conclusion	57

### Chapitre IV : Optimisation multi-objectifs du robot Delta

IV.1.	Introduction	58
IV.2.	Description de la <i>Toolbox GEA</i> sous MATLAB	58
IV.2.1.	Présentation de la structure de GEATbx	58
IV.2.2.	notations conventionnelles	59
IV.2.3.	L'arbre d'appel de GEATbx	59
IV.2.4.	Démo/ fonction de démarrage	60
IV.2.5.	Les fonctions de la toolbox (les algorithmes prédéfinis)	60
IV.2.6.	Algorithmes évolutionniste – fonctions principales	61
IV.2.6.1.	Initialisation	61
IV.2.6.2.	La boucle générationnelle des EA	61
IV.2.6.3.	La sélection	62
IV.2.6.4.	Croisement	62
IV.2.6.5.	Mutation	63
IV.2.6.6.	Evaluation	63
IV.2.6.7.	Réinsertion	63

IV.2.6.8. Migration .....	64
IV.2.6.9. Compétition.....	64
IV.2.6.10. La visualisation.....	64
IV.2.6.11. Les fonctions auxiliaires .....	64
IV.3. Formulation du problème multi-objectif : .....	64
IV.3.1. Les fonctions objectives : .....	64
IV.3.1.1. Accélération de la nacelle: .....	65
IV.3.1.2. Conditionnement de la matrice Jacobienne : .....	65
IV.3.2. Variables et contraintes : .....	65
IV.3.2.1. Limite de l'espace de travail et variables de conception.....	65
IV.3.2.2. La rigidité des barres : .....	66
IV.3.2.3. L'absence de la configuration singulière : .....	66
IV.3.2.4. Forces des Moteur : .....	67
IV.4. Procédure algorithmique développées .....	67
IV.4.1. Les fonctions MATLAB : .....	69
IV.4.2. Paramétrage : .....	69
IV.4.3. Données de l'application : .....	69
IV.4.3.1. Données de la méthode de balayage : .....	69
IV.4.3.2. Données de l'algorithme NSGA II : .....	70
IV.4.3.3. Caractéristique du robot : .....	70
IV.5. Résultats et discussions .....	70
IV.6. Conclusion.....	73

### **Chapitre V : Simulation numérique**

V.1. Introduction.....	75
V.2. Structure optimal retenue.....	75
V.3. Simulation du Robot Delta .....	76
V.3.1. Condition de la simulation .....	76
V.3.2. Simulation du Robot Delta sous MATLAB.....	76
V.3.3. Simulation par SolidWorks (COSMOS Work) : .....	82
V.3.3.1. Présentation des pièces de robot sous SolidWorks.....	82
V.3.3.2. Les résultats par COSMOS Works : .....	83
<b>Conclusion générale et perspective</b> .....	90
<b>Références Bibliographiques</b> .....	91
<b>Annexes</b> .....	95

# Liste des figures

---

Figure I- 1 : Plate-forme de Gough.....	5
Figure I- 2 : Robot parallèle Fanuc F-100.....	5
Figure I- 3 : Plate-forme de Stewart.....	6
Figure I- 4 : Prototype « main gauche ».....	6
Figure I- 5 : Photographie du robot Delta.....	7
Figure I- 6 : ABB FlexPicker - Robot Delta avec rotation de l'organe terminal.....	8
Figure I- 7 : Robot Delta à actionneurs linéaires.....	8
Figure I- 8 : Robot Hexa.....	8
Figure I- 9 : Machine Variax.....	9
Figure I- 10 : Machine Ingersoll H0h600.....	9
Figure I- 11 : Okuma Cosmo Center PM-600.....	10
Figure I- 12 : Machine HEXACT.....	11
Figure I- 13 : Hexapode CMW 300.....	11
Figure I- 14 : Hexel Tornado 2000.....	12
Figure I- 15 : Machine Toyoda HexaM.....	12
Figure I- 16 : Machine Hexaglide.....	13
Figure I- 17 : Machine Triaglide.....	13
Figure I- 18 : Renault automation Urane SX.....	14
Figure I- 19 : Machine Dyna-M.....	14
Figure II- 1 : Exemple de dominance.....	24
Figure II- 2 : Exemples de frontière de Pareto.....	25
Figure II- 3 : Sélection parallèle dans l'algorithme VEGA.....	28
Figure II- 4 : Classification des méthodes d'optimisation multi-objectifs.....	29
Figure II- 5 : Principe général des algorithmes génétiques.....	31
Figure II- 6 : Croisement à 1 point.....	34
Figure II- 7 : Croisement à 2 points.....	35
Figure II- 8 : Croisement barycentrique.....	35
Figure II- 9 : Principe de l'opérateur de mutation.....	36
Figure II- 10 : Schéma de fonctionnement de NSGA.....	38
Figure II- 11 : Calcul de la distance de crowding.....	39
Figure II- 12 : L'algorithme évolutionnaire NSGA II.....	41
Figure III- 1 : Paramétrage de la géométrie de structures de type Delta linéaire.....	46
Figure III- 2 : Points de la nacelle.....	47
Figure III- 3 : Illustration des agencements étudiés.....	50
Figure III- 4 : Schéma de principe.....	51
Figure III- 5 : Paramètres géométriques.....	52
Figure III- 6 : Répartition de la masse des barres.....	55
Figure IV- 1 : Architecture en couches du GEATbx.....	58
Table IV- 1 : Notation des fonctions dans GEATbx.....	59
Figure IV- 2 : L'arbre des appels de GEATbx.....	60
Figure IV- 3 : Schéma de résolution.....	68
Figure IV- 4 : résultat graphique par GEATbx.....	71
Figure IV- 5 : Evolution du Accélération de centre de nacelle.....	72
Figure IV- 6 : Evolution du Conditionnement de la matrice Jacobienne.....	72
Figure IV- 7 : Evolution de la Déformation des barres.....	73
Figure IV- 8 : Evolution des forces des Moteurs.....	73
Figure V- 1 : Représentation de robot Delta sous MATLAB.....	75
Figure V- 2 : Représentation de robot Delta sous SolidWork.....	76
Figure V- 3 : Les différentes étapes de notre travail.....	76
Figure V- 4 : Des instants de 1 <sup>ère</sup> cas.....	77
Figure V- 5 : Trajectoire de centre de la nacelle (1 <sup>ère</sup> cas).....	78
Figure V- 6 : Vitesse de centre de la nacelle (1 <sup>ère</sup> cas).....	78
Figure V- 7 : Accélération de centre de la nacelle (1 <sup>ère</sup> cas).....	78
Figure V- 8 : Déplacements, Vitesses et Accélérations des moteurs (1 <sup>ère</sup> cas Matlab).....	79
Figure V- 9 : Des instants de 2 <sup>ème</sup> cas.....	80
Figure V- 10 : Trajectoire de centre de la nacelle (2 <sup>ème</sup> cas).....	80
Figure V- 11 : Vitesse de centre de la nacelle (2 <sup>ème</sup> cas).....	80
Figure V- 12 : Accélération de centre de la nacelle (2 <sup>ème</sup> cas).....	81
Figure V- 13 : Déplacements, Vitesses et Accélérations des moteurs (2 <sup>ème</sup> cas Matlab).....	81
Figure V- 14 : les différentes pièces de robot Delta sous SolidWorks.....	82

Figure V- 15 : Trajectoire de centre de la nacelle (COSMOS Works).....	83
Figure V- 16 : Vitesse de centre de la nacelle (COSMOS Works) .....	83
Figure V- 17 : Accélération de centre de la nacelle (COSMOS Works).....	84
Figure V- 18 : les déplacements des moteurs (COSMOS Works) .....	84
Figure V- 19 : les vitesses des moteurs (COSMOS Works).....	85
Figure V- 20 : les accélérations des moteurs (COSMOS Works) .....	85

## Liste des tableaux

---

Table IV- 1 : Notation des fonctions dans GEATbx .....	59
Table IV- 2 : données de l’algorithme NSGA II.....	70
Table IV- 3 : Caractéristique du robot .....	70

Table IV- 4 : Solution choisie.....	71
Table IV- 5 : Valeurs maximales de Fmot et Dfb.....	73

## Notations et Abréviations

---

$\mathcal{R}_b$	Repère fixe (repère de base)
$\mathcal{R}_n$	Repère mobile (attaché à la nacelle)
$L$	Longueur des barres [m]
$R$	Rayon extérieur de robot [m]

$r$	Rayon de la nacelle [m]
$q$	Cordonnés articulaires (cordonnés des actionnaires) [m]
$\dot{q}$	Vitesse linéaire des actionnaires [m/s]
$\ddot{q}$	Accélération des actionnaires [m/s <sup>2</sup> ]
$x$	Cordonnés de centre de la nacelle [m]
$\dot{x}$	Vitesse de centre de la nacelle [m/s]
$\ddot{x}$	Accélération de centre de la nacelle [m/s <sup>2</sup> ]
$J$	La matrice Jacobienne
$u$	Vecteur unitaire
$F_{mot}$	Effort de poussée des moteurs [N]
$F_{nac}$	Torseur des efforts appliqués sur la nacelle [N]
$F_{ext}$	Efforts extérieurs (principalement les efforts de coupe) appliqués à la nacelle [N]
$F_b$	Efforts à l'intérieur des barres [N]
$c$	Conditionnement de la matrice Jacobienne
$\sigma_{max}$	La valeur singulière maximale de la matrice Jacobienne
$\sigma_{min}$	La valeur singulière minimale de la matrice Jacobienne
$\Delta x$	La déformation d'une barre [m]
$S$	Surface d'une section droite de la barre [m <sup>2</sup> ]
$E$	Module d'Young du matériau [N/mm <sup>2</sup> ]

**Abréviation :**

<b>CA</b>	Cinématique Avancé
<b>CI</b>	Cinématique Inverse
<b>GEATbx</b>	Genetic and Evolutionary Algorithm Toolbox
<b>NSGA</b>	Nondominated Sorting Genetic Algorithm
<b>VEGA</b>	Vector Evaluated Genetic Algorithm
<b>MOGA</b>	Multi-Objective Genetic Algorithm
<b>NPGA</b>	Niched Pareto Genetic Algorithm
<b>SPEA</b>	Strength Pareto Evolutionary Algorithm

# Introduction générale

---

Le domaine de la machine-outil est en plein développement afin de répondre aux besoins des utilisateurs de plus en plus exigeants sur le double plan précision et qualité. L'amélioration des performances de la machines-outils a besoin d'une évolution permanente de ce secteur afin de proposer des machines de plus en plus rapides, flexibles et intégrant l'option de l'usinage à grande vitesse (UGV), qui présente de nombreux avantages par rapport à l'usinage conventionnel. Parmi ceux-ci, nous pouvons citer entre autres :

- La réduction des efforts de coupe.
- L'augmentation de la précision de la pièce usinée, respectivement minimisation des déformations des pièces usinées.
- La stabilité dimensionnelle de la pièce est conservée, car la chaleur maximale dégagée lors de la coupe est évacuée par les copeaux et n'affecte donc pas thermiquement la pièce.
- Les efforts de coupe étant réduits et la pièce n'étant pas affectée thermiquement, ce procédé permet d'usiner plus facilement des pièces minces telles que des voiles utilisées dans l'industrie aéronautique.
- L'amélioration de l'état de surface de la pièce usinée.
- La réduction de temps de coupe et l'augmentation des cadences de production des machines, et enfin
- Usinage à sec de certains matériaux.

A cet effet, de nombreux chercheurs ont exploré le domaine de la robotique pour développer une technologie adaptée aux exigences des machines d'usinage. Dans le contexte de notre travail, nous nous intéressons aux structures parallèles destinées à l'usinage à grande vitesse, dont certaines caractéristiques géométriques et mécaniques (cinématique et dynamique) seront optimisées en fonction des exigences inhérentes à l'isotropie et à la dynamique de la machine, en tenant compte des contraintes liées aux singularités de la machine et aux déformations des barres. Il s'agit d'un problème d'optimisation multi-objectif que nous nous proposons de mettre en œuvre pour trouver des modèles de structures optimales répondant aux exigences imposées ci-dessus.

En effet, la plupart des applications traitées intègrent plusieurs critères simultanés, souvent contradictoires. En considérant par exemple deux critères contradictoires **a** et **b**, améliorer **a** détériore forcément **b** et inversement. Le concept de solution optimale devient alors plus difficile à définir. Dans ce cas, la solution optimale recherchée n'est plus un point unique, mais un ensemble de compromis. Résoudre un problème comprenant plusieurs critères, appelé communément problème multi-objectifs, consiste donc à calculer le meilleur ensemble de solutions de compromis ou «frontières de Pareto», par application de méthodes d'optimisation dites «multi-objectif». Parmi les

méthodes ayant connu un grand succès dans le domaine de la recherche-développement, nous citerons la famille des algorithmes génétiques, dont la version NSGA-II sera mise en œuvre et implémentée sous Matlab pour optimiser une structure du robot Delta.

Pour atteindre cet objectif, nous présentons notre contribution, organisée dans ce présent mémoire en cinq chapitres :

Le premier chapitre dresse un état de l'art non exhaustif des robots parallèles, et une présentation de quelques machines-outils parallèles utilisées dans l'industrie, pour mettre en exergue les grands problèmes posés dans le secteur des robots parallèles.

Le deuxième chapitre présente une synthèse des méthodes de l'optimisation multi-objectif, en introduisant des concepts fondamentaux ainsi que les méthodes de résolution utilisées. Nous décrivons les principales approches non Pareto de résolution pour ces problèmes et quelques approches Pareto, dont l'approche par algorithmes génétiques en général et par NSGA II en particulier.

Le troisième chapitre est dédié à la modélisation géométrique, cinématique, et dynamique du robot considéré. Nous choisissons les architectures parallèles candidates pour les machines-outils. Parmi ces architectures, nous prenons le robot Delta afin de concevoir une machine d'usinage trois axes, c'est-à-dire qui possède uniquement trois translations de l'outil de coupe par rapport à la pièce.

Dans le quatrième chapitre, nous présentons brièvement les outils de développement sous Matlab de l'algorithme génétique NSGA II, ou « Toolbox GEA », pour la résolution du problème posé. Les résultats obtenus sont ensuite interprétés et discutés.

Le cinquième chapitre est réservé à une simulation du modèle de robot optimisé sous Matlab et Solidworks, pour différentes configurations paramétriques, tels que les déplacements des moteurs, leurs vitesses et leurs accélérations avec un déplacement donné de la nacelle.

Une conclusion générale et des perspectives d'amélioration parachèvent ce travail.

# Chapitre I

---

## Robots parallèles pour Machines outils

---

### 1.1. Introduction

La plupart des machines-outils industriels ont une architecture cinématique série, c'est-à-dire que leur structure mobile est une chaîne ouverte formée d'une succession de segments reliés entre eux par des liaisons à un degré de liberté.

Ces robots ont pour avantage de disposer d'un grand volume de travail et d'être relativement simples sur le plan des calculs liés à leur commande.

Ces architectures ont certains inconvénients dans le domaine de l'usinage à grande vitesse :

- Les masses distribuées sur toute la chaîne cinématique résultent une inertie élevée;
- Manque de rigidité par la mise en série d'éléments élastiques ;
- Fatigue et usure des liaisons de puissance assurant l'alimentation des actionneurs ;
- Fatigue et usure des liaisons de puissance assurant la circulation des informations entre les capteurs et la commande.

A cause de ces inconvénients, les chercheurs et les entreprises commencent à étudier les machines à cinématique parallèle, parce qu'elles présentent plusieurs avantages par rapport à leurs homologues à cinématique série, comme un bon comportement dynamique (grande rigidité), une grande précision et un bon rapport entre la masse totale et la masse manipulée.

### 1.2. Définitions

Il est nécessaire de commencer par préciser la terminologie utilisée. Jean-Pierre Merlet donne les définitions suivantes [1]:

- **Un manipulateur parallèle généralisé** est un mécanisme en chaîne cinématique fermée dont l'organe terminal est relié à la base par plusieurs chaînes cinématiques indépendantes.

- **Un manipulateur parallèle** est constitué d'un organe terminal à  $n$  degrés de liberté et d'une base fixe, reliés entre eux par au moins deux chaînes cinématiques indépendantes, la motorisation s'effectuant par  $n$  actionneurs simples.

- **Un manipulateur pleinement parallèle** est un manipulateur parallèle dont le nombre de chaînes est strictement égal au nombre de degrés de liberté de l'organe terminal.

Cette terminologie constitue en elle-même les bases d'une classification.

D'autres classifications sont possibles suivant :

- le nombre de degrés de liberté de la nacelle.
- le type des degrés de liberté de la nacelle.
- les géométries de la base et de la nacelle.
- que la structure comporte ou non une ou plusieurs chaînes passives.
- que les actionneurs sont mobiles ou fixes.

Nous n'avons pas ici la prétention de dresser la liste exhaustive de tous les robots parallèles existants. Nous allons seulement présenter ceux dont l'architecture est la mieux adaptée à la réalisation de machines-outils que ce soit par la nature et le nombre des degrés de liberté de la nacelle, pour leur simplicité de réalisation (donc leur robustesse) ou pour leur bonne dynamique.

#### ➤ **Représentations utilisés**

Nous avons utilisé, pour la suite de ce manuscrit, la combinaison de deux représentations (Annexe A) :

- **Une photographie**, une vue en perspective, ou un schéma cinématique afin de visualiser facilement la disposition des liaisons.

- **Un graphe d'agencement** qui permet une compréhension aisée de la cinématique de la machine, un comptage simple des degrés de liberté de l'organe terminal et une identification de la famille à laquelle appartient cette machine.

### **1.3. Présentation de quelques robots parallèles**

Historiquement, le premier mécanisme parallèle a été imaginé par Gough [7] afin de tester le comportement de pneumatiques. La plate-forme mobile de ce mécanisme

parallèle possède 6 degrés de liberté. Le premier prototype (Figure I-1) a été réalisé en 1955.

La nacelle est reliée à la base à l'aide de 6 pattes identiques. Chacune de ces pattes est connectée, d'une part à la base par un joint de cardan et d'autre part à la nacelle par une liaison rotule. La longueur de chacune des pattes est modifiée à l'aide d'un vérin. Plusieurs robots reposant sur ce principe ont été proposés. Les variations par rapport à la plate-forme originale sont liées à la position des points d'ancrage des pattes sur la base et sur la nacelle. Ce type d'architecture est toujours d'actualité et nous pouvons citer, par exemple, les robots F-100 et F-200i actuellement commercialisés par la société Fanuc (Figure I-2).

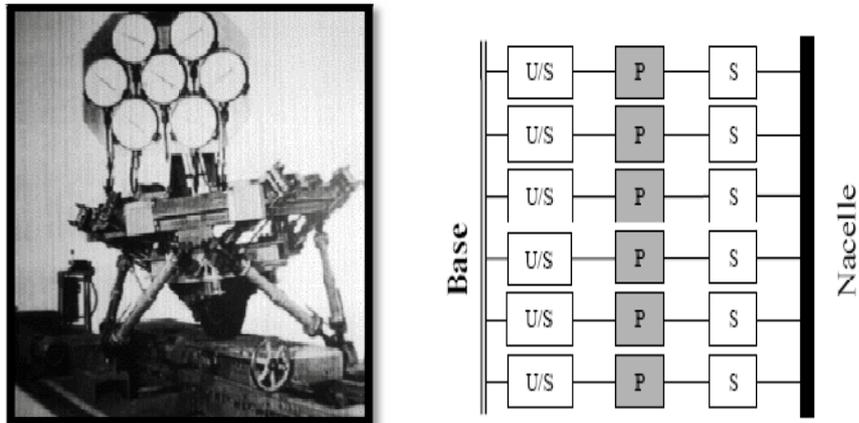


Figure I- 1 : Plate-forme de Gough. [34]



Figure I- 2 : Robot parallèle Fanuc F-100. [37]

Ce principe, légèrement modifié (Figure I-3), a été repris en 1965 par Stewart [8] pour réaliser un simulateur de vol à 6 degrés de liberté. Cependant, de par l'architecture retenue, nous constatons que l'objectif de la réduction des masses en mouvement n'est pas pleinement atteint. Les plates-formes de Gough et de Stewart, souvent appelées abusivement toutes deux « plate-forme de Stewart », possèdent des actionneurs mobiles et par conséquent une dynamique réduite, bien que déjà meilleure que celle des robots série.

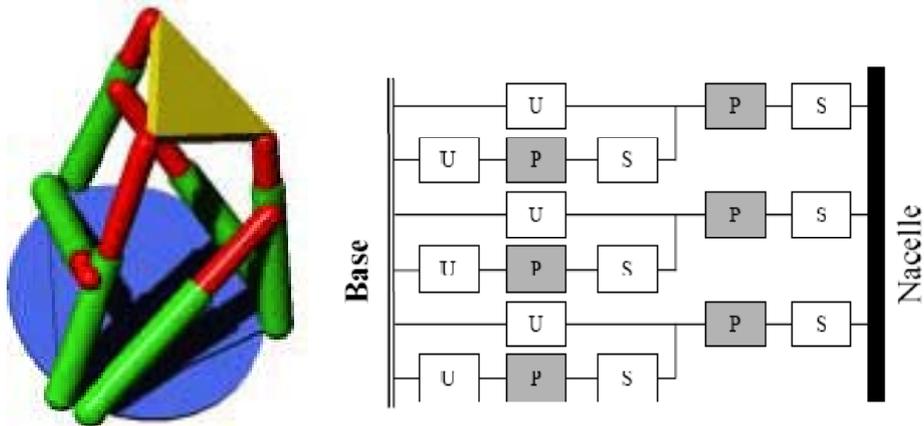


Figure I-3 : Plate-forme de Stewart. [35]

Pour améliorer la dynamique de la plate-forme de Gough, l'idée est d'invertir les liaisons glissière et cardan en utilisant des barres de longueur fixe donc plus légères. Les actionneurs sont alors disposés sur la base. Cela aboutit au prototype « main gauche » développé à l'INRIA par Jean-Pierre Merlet [2]. Ce robot, dont la plate-forme mobile possède également 6 degrés de liberté (Figure I-4), est constitué de 6 actionneurs linéaires reliés chacun à une barre de longueur fixe par un joint de cardan. La deuxième extrémité de ces barres est en liaison rotule par rapport à la nacelle.

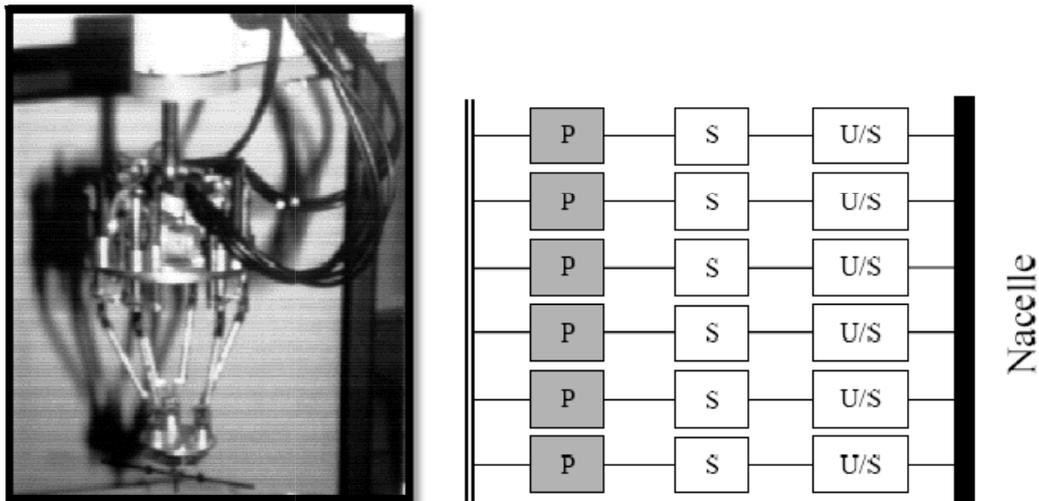


Figure I-4 : Prototype « main gauche »

L'organe terminal (nacelle) des robots que nous venons de présenter possède 6 degrés de liberté (3 translations et 3 rotations).

Le robot Delta, créé à l'Ecole Polytechnique Fédérale de Lausanne [9]. Les mouvements de l'organe terminal de ce robot sont les 3 translations. La constitution de la première version du Delta est représentée sur la Figure I-5



*Figure I- 5 : Photographie du robot Delta*

L'apparition de cette structure ultralégère a apporté un renouveau au domaine de la robotique parallèle. Plusieurs caractéristiques de ce robot sont :

- La motorisation de ce robot peut être réalisée à l'aide d'actionneurs linéaires (Figure I-7) ou rotatifs (Figure I-5).
- Les barres reliant les moteurs à la nacelle peuvent être du type :
  - Doubles barres de même longueur montées sur rotule à chacune de leurs extrémités (ou cardan à une extrémité, rotule à l'autre). Si les barres d'une paire n'ont pas la même longueur, les mouvements de la nacelle ne sont pas des translations pures.
  - Simples barres possédant une liaison cardan à chacune de leurs extrémités. Dans ce cas, une condition géométrique régit la position des axes des cardans afin d'obtenir les mouvements de translation de la nacelle. De la même manière, si cette condition géométrique n'est pas respectée, les mouvements de la nacelle ne sont pas des translations pures. Par exemple, le robot Georg V de l'IFW (université d'Hanovre) ne respecte pas ces conditions. Sa nacelle ne se déplace donc pas en translation, cependant, pour ce robot, cela n'est pas pénalisant car la nacelle porte un poignet 2 axes.
  - Pour les architectures à base d'actionneurs rotatifs, on peut ajouter un mouvement de rotation de l'organe terminal (Figure I-6). Dans ce cas, l'actionneur est localisé sur la base afin de ne pas pénaliser la dynamique de l'ensemble. Le mouvement est transmis à l'organe terminal par une chaîne passive.

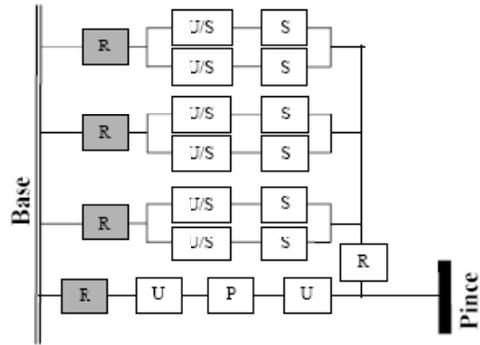


Figure I- 6 : ABB FlexPicker - Robot Delta avec rotation de l'organe terminal.

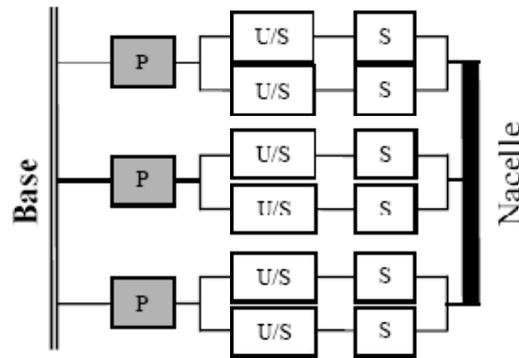
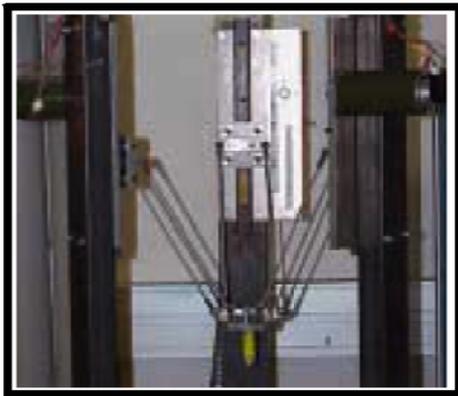


Figure I- 7 : Robot Delta à actionneurs linéaires.

En1991, François Pierrot propose une évolution du concept 3 axes du robot Delta vers un concept 6 axes. Ses travaux ont débouché sur la création du robot Hexa représenté sur la Figure I-8. Il est composé de 6 actionneurs rotatifs reliés chacun par une liaison rotule à une barre de longueur fixe. Chacune de ces barres est reliée à la nacelle par une liaison rotule. Tout comme le robot Delta, il s'agit d'un robot léger, donc rapide.

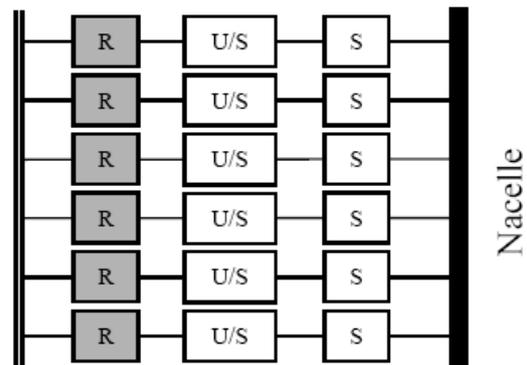
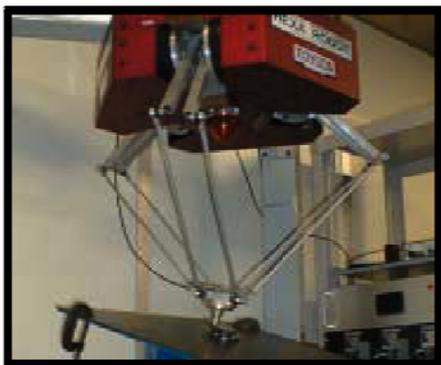
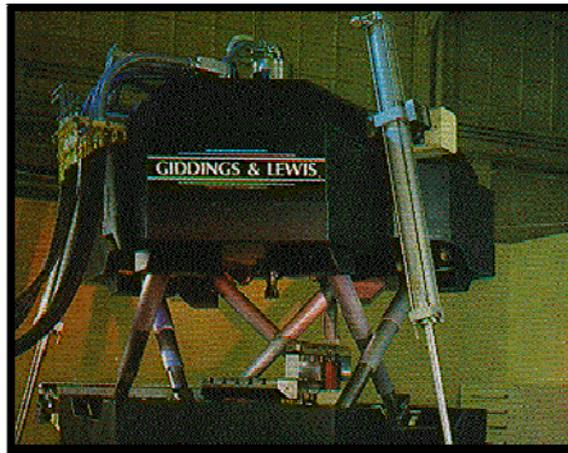


Figure I- 8 : Robot Hexa.

#### 1.4. Machines-outils parallèles

La première machine d'usinage à architecture parallèle proposée est la Variax de Giddings & Lewis (Figure I-9) en 1994 [10]. Sa structure parallèle est du même type que celle de la plate-forme de Gough. Nous appellerons les machines de ce type «Hexapodes».



*Figure I- 9 : Machine Variax.*

Beaucoup de machines-outils de la famille des hexapodes ont été proposées. Parmi les plus connues nous trouvons :

#### **1.4.1. La machine Ingersoll H0H600.**

Il s'agit d'une des premières machines d'usinage 5 axes du type hexapode. 6 pattes télescopiques montées sur rotules relie la nacelle au bâti. La version présentée ici est à broche horizontale. Une version à broche verticale existe également.



*Figure I- 10 : Machine Ingersoll H0h600.*

#### **1.4.2. La machine Mikromat 6X Hexapode.**

Comme son nom l'indique cette machine appartient à la famille des hexapodes. Elle est similaire à la machine INGERSOLL ci-dessus. Sa principale différence est la disposition des articulations rotule sur la nacelle. En effet, les centres des liaisons rotule n'appartiennent pas au même plan. Ils sont situés dans deux plans distincts perpendiculaires à l'axe de la broche. Cette disposition constructive est intéressante afin de réduire les risques de collision entre barres. Par contre, pour les pattes les plus éloignées de l'outil, le débattement angulaire est accentué ce qui contribue à réduire la plage de variation de l'orientation de la nacelle.

#### 1.4.3. La machine Cosmo Center PM-600 de la société Okuma.

Il s'agit également d'une machine de type hexapode développée par la société Okuma.



Figure I- 11 : Okuma Cosmo Center PM-600.

#### 1.4.4. La machine HEXACT développée par INA et l'IFW.

Il s'agit encore d'une machine de type hexapode dont la disposition des pattes est intéressante. En effet, comme la machine Mikromat, les pattes sont connectées à la nacelle dans deux plans perpendiculaires à l'axe de la broche. Ces plans sont situés de part et d'autre d'un plan médian sur lequel la deuxième extrémité des pattes est connectée. Les six points de connexion sur le plan médian constituent les sommets d'un hexagone régulier. Cette disposition des liaisons est intéressante du point de vue de la simplicité de conception de la machine et de sa compacité. Par contre, elle souffre de deux défauts majeurs :

- Comme pour la machine Mikromat, les débattements angulaires sont réduits à cause de l'éloignement des plans de connexion par rapport à l'outil.

- La machine, telle qu'elle a été conçue, présente une singularité au centre de son volume de travail (raideur nulle pour un moment appliqué autour de l'axe de la broche quand cette dernière a une orientation nulle).

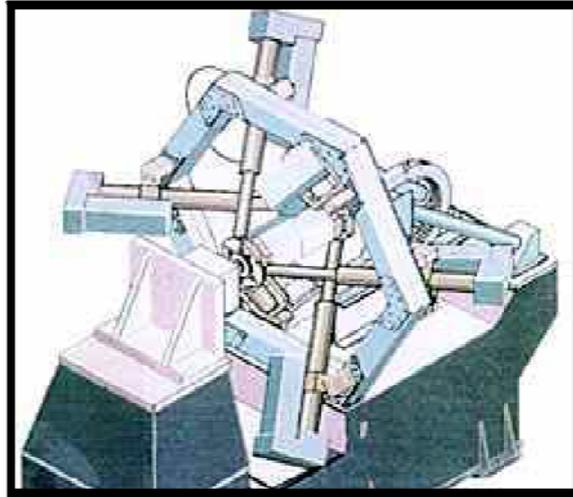


Figure I- 12 : Machine HEXACT.

#### 1.4.5. L'Hexapode 300 de la société CMW.

Cette machine (Figure 1-13) est de type hexapode. Elle peut être utilisée soit de manière autonome (son volume de travail est alors limité), soit portée par une structure cartésienne destinée à la positionner en plusieurs endroits afin de recouvrir un volume de travail plus important.

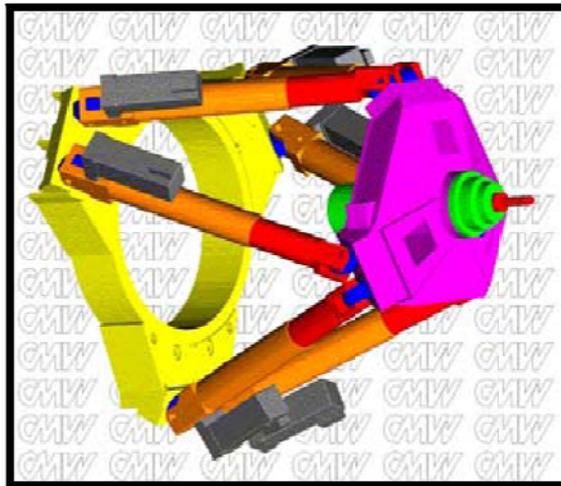


Figure I- 13 : Hexapode CMW 300. [34]

#### 1.4.6. La machine Tornado 2000 de la société Hexel.

Il s'agit d'une machine de type hexapode similaire à la machine INGERSOLL.



Figure I- 14 : Hexel Tornado 2000.

Les machines ci-dessus étant de la famille des hexapodes, ce sont toutes des machines 5 axes au sens de l'usinage. Nous avons vu que la dynamique des robots parallèles est meilleure lorsque les actionneurs sont fixés sur la base. Les concepteurs de machine-outil ont donc proposé des solutions de machines, inspirées des robots parallèles correspondants:

#### 1.4.7. La machine HexaM de la société Toyoda.

Cette machine construite par la société Toyoda est une variante du robot Hexa. Il s'agit en fait de la déclinaison du robot Hexa avec des actionneurs linéaires.

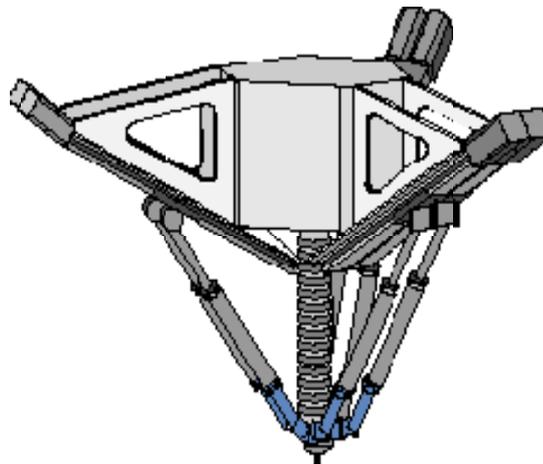


Figure I- 15 : Machine Toyoda HexaM. [34]

#### 1.4.8. La machine Hexaglide de l'ETHH (Zurich).

L'Hexaglide est une machine qui appartient à la famille « main gauche ». Elle est composée de 6 actionneurs linéaires fixes et d'une nacelle reliés par des barres de longueur fixe montées sur rotule. Il s'agit d'une machine d'usinage 5 axes. Sa

particularité réside dans la disposition des liaisons glissières qui sont toutes coplanaires et de même direction. Cette disposition permet d'avoir une direction de déplacement privilégiée afin d'usiner des pièces longues.



Figure I- 16 : Machine Hexaglide

#### 1.4.9. La machine Triaglide

Elle ne possède que trois degrés de liberté de translation. La machine Triaglide est une machine 3 axes à actionneurs linéaires. Il s'agit de la version machine-outil de l'architecture du robot Delta. Les liaisons glissières sont parallèles entre elles ce qui permet d'avoir une direction de déplacement privilégiée pour l'usinage de pièces longues.

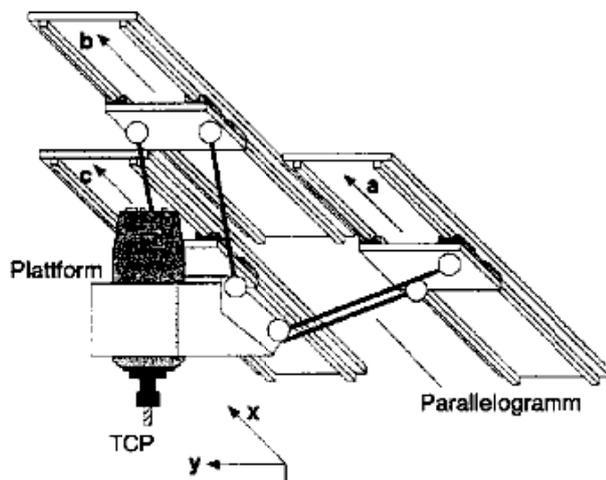
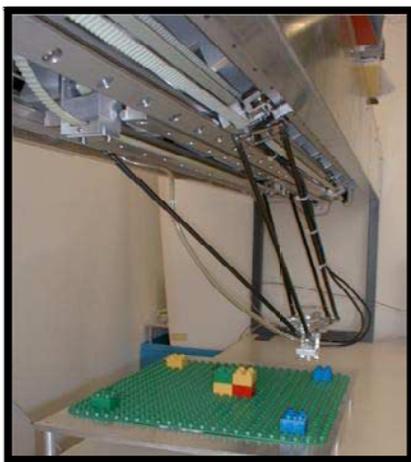


Figure I- 17 : Machine Triaglide.

#### 1.4.10. La machine UraneSX de la société Renault Automation/Comau

Qui possède trois degrés de translation. Cette machine, équipée de moteurs linéaires, est capable d'accélérer à 3,5g sur l'ensemble de son volume de travail. Elle a été conçue

afin de réaliser spécifiquement des opérations de perçage, lamage et taraudage à grande vitesse [11].

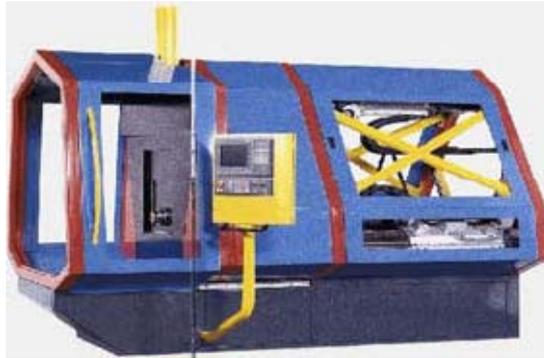


Figure I- 18 : Renault automation Urane SX [34]

Il existe également des machines hybrides ayant une ou plusieurs des caractéristiques suivantes :

- Seul un sous-ensemble de la machine est parallèle.
- La machine n'est pas pleinement parallèle (les pattes de la machine comportent plusieurs actionneurs en série).
- La machine comporte une chaîne passive.

Parmi ces machines hybrides, nous trouvons :

#### 1.4.11. La machine Dyna-M du WZL (Aix-la-Chapelle)

Il s'agit d'une machine 3 axes constituée d'un mécanisme plan à deux degrés de liberté complété par un axe linéaire porté.

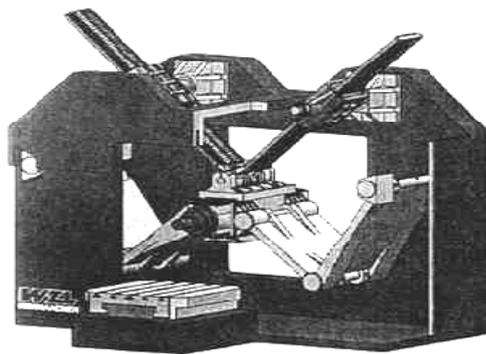


Figure I- 19: Machine Dyna-M.

### 1.5. La problématique de mécanisme parallèle

#### 1.5.1. Introduction

D'après les études continues dans les laboratoires sur les robots parallèles, ces types de robots sont maintenant utilisés dans plusieurs applications réelles telles que les applications médicales, divertissements et le domaine de la machine outil. Cet intérêt provient des avantages des mécanismes parallèles, les plus notables : la haute précision, la rigidité, la vitesse et la grande capacité de charge. Lesquels dans un très grand nombre de cas peuvent vaincre les inconvénients de la cinématique complexe et l'espace de travail petit. Dans cette partie, nous allons examiner certaines questions clés dans ce domaine.

### 1.5.2. Les différentes couches d'un système de robot parallèle

Comme leur équivalent série les robots parallèles sont constitués de plusieurs couches.

#### a) La couche du mécanisme

C'est le robot lui-même avec un premier modèle théorique constitué de :

- **la topologie du mécanisme** c.-à-d. comment les articulations, liaison et actionneurs sont arrangés pour produire le mouvement souhaité.
- **la géométrie du mécanisme** c.-à-d. les dimensions des liaisons, l'emplacement des articulations,

#### b) La couche du contrôle

Elle est constituée de:

- un niveau de communication qui permet le transfert d'informations entre les capteurs et les actionneurs du robot et le contrôleur.
- Un niveau de contrôle qui peut être décomposé:
  - Un niveau *d'organisation de mouvement* qui génère une séquence de mouvements pour le robot.
  - un *contrôleur* qui assure l'exécution des mouvements élaboré par le planificateur du mouvement.

#### c) La couche de La conception et de la simulation

Est constituée de :

- un *module de conception* qui permet de déterminer la topologie et la géométrie théorique qui sont les mieux adaptées pour les tâches à effectuer par le robot.

- un *module de simulation* qui permet de simuler le comportement d'un robot de topologie et de géométrie données.

Comme une option nous pouvons avoir aussi une couche de calibrage dont le but est d'obtenir une meilleure adéquation entre le modèle théorique de la géométrie et sa géométrie réelle en utilisant les capteurs du robot ou des capteurs supplémentaires.

### 1.5.3. Le Mécanisme

#### a) Les architectures Mécaniques

Plus de 100 architectures mécaniques différentes de robots parallèles ont déjà été proposées et il est probable que toutes n'ont pas été découvertes. Malheureusement, il n'y a pas beaucoup d'architectures proposées qui ont seulement 4 ou 5 ddl. Tandis que de nombreuses applications nécessitent un tel nombre de ddl. D'où une tendance récente de proposer des robots parallèles avec moins de 6 ddl [3].

Il s'agit clairement d'un domaine de recherche intéressant mais de nombreuses questions se posent avec ce type de robots:

- la structure proposée a, en théorie, seulement 4 ou 5 ddl et de s'appuyer sur les contraintes géométriques pour obtenir cette réduction du nombre de ddl. Toutefois, dans la pratique, ces contraintes ne seront jamais parfaitement accomplies, et par conséquent ces robots présenteront des mouvements parasites. Le problème consiste donc à déterminer quel sera l'amplitude maximum de ce mouvement parasite, qui déterminera l'amplitude des tolérances de fabrication [12], afin que l'amplitude maximale du mouvement parasite ne puisse pas dépasser un seuil donné.

- bien que présentant moins de capteurs et d'actionneurs, cela peut paraître une solution économiquement intéressante. En effet en premier lieu toutes leurs chaînes cinématiques sont identiques (ce qui n'est pas le cas pour un robot de 4 et 5 ddl) qui permettra de réduire les coûts d'entretien. Ensuite, en utilisant la redondance, il est possible d'optimiser les performances du robot pour une tâche donnée : par exemple pour les opérations d'usinage qui nécessitent seulement 5 ddl, il est possible d'utiliser les ddl supplémentaires de la plate-forme de Gough (la rotation de la plate-forme autour de sa propre normale), afin que la raideur totale sur une trajectoire typique soit de 5 à 25% plus grande que la raideur d'un robot identique dans lequel la redondance n'est pas utilisée [13].

#### b) Articulation, capteurs et actionneurs

Les robots parallèles nécessitent de plus grands couples cinématiques avec une amplitude relativement importante du mouvement, et dans certains cas, avec une haute charge. Actuellement les articulations disponibles (à rotule ou U-Joints) ne sont pas complètement satisfaisantes de ce point de vue, bien que ce soit des produits récents, comme l'INA ou les articulations de Hephaist ont été développés spécialement pour les

robots parallèles [16]. Donc le développement de grands couples cinématiques avec 2 à 4 ddl est une question clé. Comme pour toutes les articulations mécaniques, ces articulations doivent avoir un faible coefficient de frottement, sans jeu. Mais, en outre, ces articulations doivent être conçues de sorte qu'il est possible d'ajouter des capteurs pour mesurer en partie ou totalement, l'amplitude du mouvement de l'articulation (ce qui est important pour la cinématique avancée tel que mentionné dans la section suivante). Notez aussi que la flexibilité des articulations est également un domaine de recherche intéressant, en particulier pour les micro-robots [11]. En ce qui concerne les actionneurs beaucoup de robots utilisent des actionneurs linéaires. Dans le domaine des machines-outils quelques robots parallèles tel que l'Urane SX d'Automatisation Renault utilise le moteur électrique linéaire qui présente des accélérations impressionnantes. Mais il ya un manque de capteurs et d'actionneurs linéaires pour les micro-robots parallèles [2].

#### 1.5.4. Cinématique

##### a) Cinématique Inverse

Tout le monde consent que la cinématique inverse soit un élément de base pour n'importe quel contrôleur de robot. Heureusement, il est connu que la cinématique inverse est habituellement facile pour tout robot parallèle. Plus précisément, dans la plupart des cas :

- il existe une solution unique à la cinématique inverse (dans certains cas pourvu que les contraintes physiques soient prises en compte comme pour le robot Delta [14]).
- Chaque variable articulaire peut être calculée de façon indépendante, étant donné la position désirée du robot.

Ce dernier point est la différence principale avec le robot série et permet une cinématique inverse très rapide à condition que le matériel du contrôleur soit approprié.

##### b) Cinématique avancée

Le problème majeur de la cinématique est la cinématique avancée qui consiste à trouver la possibilité de positionnement de la plate-forme pour des coordonnées d'articulation données. La cinématique avancée est un problème plus complexe que son équivalent la cinématique inverse pour un robot série. La nécessité de la *CA* est une question controversée. On peut penser que la *CA* est une question académique qui peut être seulement utile hors ligne à des fins de simulation d'un robot parallèle pour lequel ne sera contrôlée que la position en utilisant seulement la *CI*.

#### 1.5.5. Singularités

Il ya plusieurs façons d'introduire la notion de singularités, mais la plus spectaculaire est de considérer le comportement statique du robot. Soit  $\mathcal{F}$  le torseur appliqué sur la

plate-forme du robot et  $\tau$  l'ensemble des forces articulaires. Ces quantités sont linéairement liées par :

$$\mathcal{F} = \mathbf{J}^{-T} \tau$$

Où  $\mathbf{J}^{-T}$  est la transposé de la matrice inverse du Jacobienne du robot. Chaque composante du vecteur des forces articulaires peuvent ainsi être obtenus sous forme de ratio:

$$\tau_i = \frac{A}{|\mathbf{J}^{-T}|}$$

Où  $A$  est le mineur associé  $\tau_i$  à, Par conséquent à condition que  $A$  ne soit pas égale à 0, la force articulaire  $\tau_i$  tendra vers l'infini en n'importe quelle position, appelées positions singulières où le déterminant est égale à zéro, ce qui provoque une panne du robot.

Bien que la condition  $|\mathbf{J}^{-T}| = 0$  paraisse être une condition simple comme la matrice  $\mathbf{J}^{-T}$  a une forme analytique, le calcul complet de ce facteur conduit à une expression complexe avec un grand nombre de termes (en particulier si le robot a 6 ddl), ce qui est inutile dans la pratique.

Nous avons maintenant une meilleure compréhension des configurations singulières. Ils se produiront pour des configurations géométriques spécifiques du robot qui peuvent être déterminés, quel que soit le nombre de ddl du robot. Nous disposons maintenant d'un algorithme efficace qui permet de déterminer s'il existe des configurations singulières soit dans un espace de travail accessible ou dans un espace de travail spécifique pour la plate-forme [4]. Nous pouvons également tester en temps quasi réel la présence de la singularité sur une trajectoire arbitraire [5].

Mais cela ne signifie pas que tous les problèmes liés aux configurations singulières sont résolus. Par exemple une meilleure caractérisation de la configuration singulière est exigée. En effet la singularité est dangereuse si seulement le dénominateur de  $\tau_i$  tend vers 0. En effet, si le numérateur aussi tend vers 0, alors la force articulaire peut être encore finie.

### 1.5.6. Espace de travail

Il est bien connu que le principal inconvénient des robots parallèles est leurs espaces de travail réduit. En outre le calcul de cet espace de travail n'est pas une tâche facile car, à l'opposé du robot de série classique, l'espace de travail de translation et d'orientation sont couplés. Classiquement, une première approche pour résoudre ce problème est de fixer les valeurs de certaines ddl, en principe on peut aller seulement jusqu'à 3 ddl. Cela se fait généralement soit par la fixation de l'orientation de la plate-forme ou l'emplacement de son centre. Dans le premier cas, l'approche géométrique qui détermine

géométriquement le mouvement possible du centre de la plate-forme pour chaque chaîne cinématique entraîne généralement à un meilleur résultat comme il fournit un calcul exact avec un stockage compact et une représentation facile [17]. L'orientation de l'espace de travail est plus difficile à traiter car il n'existe pas de moyen universel pour représenter cet espace de travail.

Une autre approche consiste à calculer une approximation soit de la frontière ou de l'espace de travail entier en utilisant une méthode numérique [6]. Certaines de ces approches ont l'avantage de donner la possibilité de traiter également des limites sur le mouvement des articulations passives et de permettre la vérification de l'espace de travail (c.-à-d. vérifier si un espace de travail désiré est inclus dans l'espace de travail du robot). Ils peuvent également calculer les différents types d'espace de travail (par exemple pour déterminer tous les emplacements possibles du centre de la plate-forme telle qu'il est possible d'avoir une orientation de la plate-forme au sein de certaines limites prescrites pour les angles d'orientation).

Dans ce domaine reste deux problèmes non résolus:

- un algorithme rapide pour calculer le mouvement maximal de la plate-forme
- Un algorithme qui permet de vérifier les liens d'interférence. C'est un problème beaucoup plus complexe qui peut être envisagé. En effet il est nécessaire de déterminer toutes les hyper-surfaces dans l'espace de travail pour lequel une paire de chaîne de la cinématique se croise dans le but de diviser l'espace de travail dans les régions sans interférences. C'est une tâche difficile, même pour robots avec chaînes cinématiques très simple.

### 1.5.7. Organisation du mouvement

L'Organisation du mouvement est un problème classique pour robot série. Mais dans le cas de robots parallèles le problème est différent quelque peu. Les problèmes possibles sont:

1. Vérifier si une trajectoire donnée est complètement dans l'espace de travail du robot.
2. déterminez si deux positions peuvent être atteintes par une singularité et la trajectoire sans interférence qui se trouve entièrement dans l'espace de travail du robot.

Le problème 1 peut être résolu pour presque n'importe quelle trajectoire arbitraire en utilisant l'analyse d'intervalle [5], (que) le problème 2 n'a aucune solution connue à ce temps.

### 1.5.8. Étalonnage

L'étalonnage est un problème bien connu pour les robots séries et est maintenant un problème bien traité. L'étalonnage de robots parallèles peut s'appuyer sur les méthodes développées pour le robot série mais malheureusement ce n'est pas exactement le cas. En effet il y a une différence majeure entre les deux robots: pour les robots série les petites erreurs sur les paramètres géométriques induisent de grandes erreurs sur le positionnement de l'organe terminal pendant que pour les robots parallèles ces erreurs seront petites. La Simulation pour l'étalonnage est essentielle: elle permet de déterminer combien les méthodes d'étalonnages sont sensibles au bruit dans les mesures et aux erreurs numériques.

Il existe deux types de méthodes d'étalonnage:

- **externe**: un appareil de mesure externe est utilisé pour déterminer (entièrement ou partiellement) quelle est la position réelle de la plate-forme pour les différentes configurations désirées. Les différences entre la position mesurée et la position désirée donnent un signal d'erreur qui est utilisé pour l'étalonnage [15].
- **auto-étalonnage**: la plate-forme a des capteurs supplémentaires (par exemple des capteurs qui sont utilisés pour la CA) et seuls les mesures du robot sont utilisées pour le l'étalonnage [18].

La première méthode est difficile et fastidieuse à utiliser dans la pratique mais peut donner des bons résultats. La deuxième méthode peut être moins exacte mais peut être facile à utiliser et peut avoir l'avantage d'être entièrement automatisée.

Un problème théorique intéressant est de déterminer quelles sont les configurations de mesure de la plate-forme qui mèneront aux meilleurs étalonnages. Alors il y a aussi le problème de mise en œuvre du calibrage dans un environnement industriel.

### 1.5.9. Dynamique

Un autre avantage des robots parallèles est qu'ils peuvent atteindre de grandes vitesses et accélérations, dû à leur faible masse mobile [12]. Mais le contrôle de tels robots est une tâche difficile: bien que les nombreux travaux aient rapporté des méthodes pour calculer le modèle dynamique d'un robot parallèle. Un problème important est de déterminer ce qui devrait être le temps du calcul du modèle dynamique de sorte que son utilisation dans une boucle de contrôle mène réellement à une amélioration des performances du robot. C'est une question très complexe, surtout s'il est considéré que le contrôle de l'algorithme n'est pas continu. La deuxième question clé dans ce domaine est la mise en œuvre et l'application du schéma de contrôle.

### 1.5.10. Synthèse

C'est bien connu que les performances qui seront atteintes par un mécanisme dépendent de:

- la *topologie* du mécanisme
- *des dimensions* des composants du mécanisme

Cela est particulièrement vrai pour les mécanismes en boucle fermée qui sont très sensibles à ces deux facteurs. D'où la conception d'un mécanisme parallèle doit être menée de sorte que ses performances correspondent le mieux à une liste d'exigences et à la fois aux différents aspects qui doivent être abordés:

- *synthèse topologique* c.-à-d. trouver les dispositions générales des articulations, et des liaisons qui permettront de décrire la cinématique générale de la structure.
- *synthèse dimensionnelle* c.-à-d. trouver le dimensionnement approprié du mécanisme.

La synthèse de robot parallèle est un domaine ouvert (il ya un nombre très limité de documents traitant cette question). L'utilisation de structures parallèles dans le domaine de la machine-outil a montré que les concepteurs ont une compréhension profonde des mécanismes en boucle ouverte, mais ont un manque total d'expérience en boucle fermée, ils se sont concentrés seulement sur le développement des composants mécaniques de base de leur machine et ont presque complètement négligé la partie de l'analyse.

### 1.5.11. Contrôleur

Les développements proposés dans les sections précédentes conduiront à un système efficace que si le contrôleur du robot permet de négocier avec les spécificités de robots parallèles. Malheureusement, la tendance actuelle, en particulier dans le domaine de la machine-outil, est d'essayer d'adapter le matériel existant dans le but de contrôler des robots parallèles. Si nous prenons comme exemple le domaine de la machine-outil nous pouvons analyser les erreurs sur les parties fabriquées qui sont dû à chaque élément du système: le système CAO qui est utilisé pour définir les parties et qui est générateur des propositions pour la plate-forme, le contrôleur qui surveille l'exécution de ces propositions et finalement la plate-forme elle-même. L'utilisation de la technologie actuelle, a pu démontrer que le système de CAO est responsable d'environ 20% des erreurs, la plate-forme (conception optimisée) moins que 10% pendant que le contrôleur induit 70% des erreurs. D'où les recherches devraient se concentrer sur le système CAO (mais les méthodes existantes peuvent améliorer déjà cette partie) et principalement le contrôleur. Le matériel du contrôleur devrait nous permettre:

- la possibilité d'utiliser des lois de contrôle appropriées, notamment le contrôle de vitesse.

- calcul parallèle (cela améliorera considérablement le temps d'échantillonnage)
- les circuits intégrés spécialisés, qui seront consacrés aux tâches de calcul tel que cinématique inverse et avancée

## 1.6. Conclusion

Dans cette partie, nous avons présenté l'histoire des robots parallèles et citer les grandes familles de ce type de robots, puis nous avons exposé les robots parallèles les plus utilisés dans le domaine de la machine-outil. En fin, nous avons tenté de souligner certains des problèmes ouverts dans le domaine des robots parallèles (sans prétention d'être exhaustive). Certains de ces problèmes sont à long terme pendant que d'autres sont des questions principales pour les possibilités d'utilisation des robots parallèles dans la pratique à court terme. Au cours des 20 dernières années, nous avons acquis une meilleure compréhension du comportement de ces mécanismes en boucle fermée complexe, mais il subsiste encore beaucoup de problèmes non résolus et passionnants. Si l'on compare ces 20 ans aux 200 ans qui ont été nécessaire pour atteindre le niveau actuel de réalisation de mécanismes séries on peut conclure qu'il ya encore un long chemin à parcourir sur la route des mécanismes parallèles.

# Chapitre II

## Méthodes d'Optimisation multi-objectifs

### II.1. introduction

La plupart des problèmes réels requièrent l'optimisation simultanée de plusieurs objectifs. Dans le cas de l'optimisation monocritère, la solution optimale est facilement définie ; ce n'est pas ainsi dans le cas de plusieurs objectifs. Au lieu d'une solution unique, le résultat d'une proposition multicritère est généralement un assortiment de solutions, qui se distinguent par différents compromis réalisés entre les objectifs. Cet assortiment est connu comme *Pareto-optimals*. Les solutions qui le composent sont optimales, dans le sens qu'il n'existe dans l'univers de recherche aucune solution meilleure si tous les objectifs sont considérés simultanément. Ainsi, le but de l'Optimisation Multi-objectif, aussi appelée Multicritère de ou Vectorielle, consiste à obtenir les solutions de Pareto et, par conséquent, à connaître l'ensemble des compromis possibles entre les objectifs. Cela fournit à l'ingénieur une meilleure compréhension de son problème.

### II.2. Définitions

#### II.2.1. Problème d'optimisation multi-objectif

D'une façon générale, un problème d'optimisation multi – objectifs est exprimé par l'équation :

$$\begin{cases} \min F(x) = (f_1(x), f_2(x), \dots, f_n(x)), x \in E = \mathfrak{R}^m \\ g_i(x) \leq 0, & i = 1, \dots, p \\ h_j(x) = 0, & j = 1, \dots, q \\ x_{k \min} \leq x_k \leq x_{k \max}, & k = 1, \dots, m \end{cases}$$

où  $n > 2$  est le nombre de fonctions objectif et  $F(x) = (f_1(x), \dots, f_n(x))$  est le vecteur des fonctions à optimiser. Le vecteur  $x = (x_1, \dots, x_m)$  est le vecteur des variables de décision. ( $g_i(x) \leq 0$ ,  $h_j(x) = 0$ ,  $x_{k \min} \leq x_k \leq x_{k \max}$ ) sont respectivement les contraintes d'inégalités, les contraintes d'égalités et les contraintes de domaine.

#### II.2.2. Solution d'un problème d'optimisation multi-objectif

Alors que dans le cas d'un problème d'optimisation mono-objectif, il n'existe qu'un seul optimum global avec éventuellement plusieurs solutions correspondantes, ce n'est plus le cas pour un problème d'optimisation multi-objectif. Il faut donc utiliser une autre notion d'optimum. La plus communément admise est l'optimalité Pareto. On parle alors d'optimum Pareto.

### i. Optimum de Pareto

Au XIX<sup>ème</sup> siècle, Vilfredo Pareto, un mathématicien italien, formule le concept suivant : dans un problème multi-objectif, il existe un équilibre tel que l'on ne peut pas améliorer un critère sans détériorer au moins un des autres critères. Cette équilibre a été appelé optimum de Pareto. Un point  $x$  est dit **Pareto-optimal** s'il n'est dominé par aucun autre point appartenant à  $E$ . Ces points sont également appelés solutions non inférieures ou non dominées.

### ii. La notion de dominance

$x \in E$  domine  $x' \in E$  si  $\forall i, f_i(x) \leq f_i(x')$  avec  $\exists i$  telque  $f_i(x) < f_i(x')$

Dans l'exemple ci-dessous, les points 1,3 et 5 ne sont dominés par aucun autre. Alors que le point 2 est dominé par le point 1, et que le point 4 est dominé par les points 3 et 5.

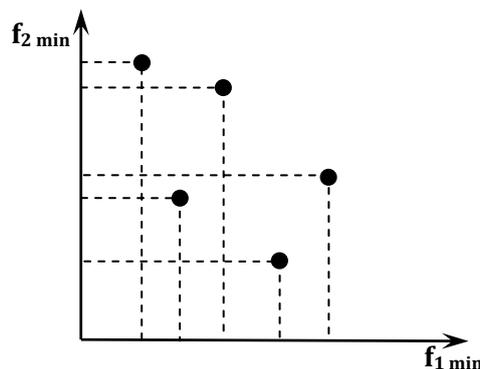


Figure II- 1 : Exemple de dominance [20]

### iii. La frontière de Pareto

La frontière de Pareto est l'ensemble de tous les points Pareto-optimaux. Les figures ci-dessous présentent pour un problème à deux objectifs les quatre frontières de Pareto en fonction du désir de l'utilisateur de minimiser ou maximiser les objectifs.

Dans l'exemple de la Figure II-3, la frontière de Pareto est composée des points 1,3 et 5.

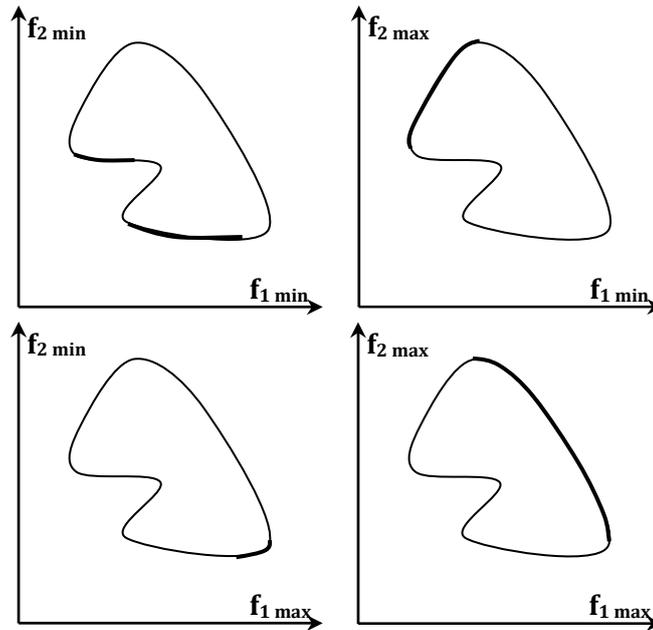


Figure II- 2 : Exemples de frontière de Pareto [20]

### II.3. Les méthodes de résolution

Dès lors résoudre un problème multiobjectifs ne consiste pas à rechercher la solution optimale, mais l'ensemble des solutions satisfaisantes, pour lesquelles on ne pourra pas effectuer une opération de classement. Dans ce cas il faut utiliser des méthodes capables de trouver l'ensemble de Pareto tout en se basant sur les contraintes et la définition de problème. Dans la littérature, il existe plusieurs méthodes pour ce but, les premières tendances aboutissent à des méthodes purement mathématiques dont elles ne peuvent pas tendre vers l'ensemble de solutions dans un temps optimal et elles sont compliquées, par exemple, la programmation linéaire, la méthode de Simplexe, ... . On outre, une deuxième tendance a pour but d'avoir des méthodes non mathématiques et tend vers l'ensemble de solutions dans un temps optimal et d'une manière simple en se basant sur le principe des heuristiques, cette tendance est « les méta-heuristiques », comme les algorithmes évolutionnaires, recuit simulé, ...

Ces méthodes de ces cinq groupes peuvent aussi être rangées en trois familles des méthodes d'optimisations multi-objectifs:

➤ **Les méthodes a priori (décideur → recherche)**

Les solutions les plus intuitives pour résoudre des problèmes multi-objectifs consistent souvent à combiner les différentes fonctions objectives en une fonction d'utilité suivant les préférences du décideur. Dans ce cas le décideur est supposé connaître a priori le poids de chaque objectif afin de les mélanger dans une fonction unique. Cela revient à résoudre un problème simple objectif.

➤ **Les méthodes a posteriori (recherche → décideur)**

Le décideur prend sa décision d'après un ensemble de solutions calculées par un solveur. Dans ce cas la qualité de la décision dépend du choix de la méthode de résolution. Car celle-ci va devoir donner un ensemble de résultats le plus représentatif de l'espace des objectifs efficaces.

➤ **Les méthodes progressives (décideur ↔ recherche)**

Dans ces méthodes, les processus de décision et d'optimisation sont alternés. Par moment, le décideur intervient de manière à modifier certaines variables ou contraintes afin de diriger le processus d'optimisation. Le décideur modifie ainsi interactivement le compromis entre ses préférences et les résultats.

Les méthodes méta-heuristiques peuvent classer en trois catégories, des méthodes agrégées, des méthodes Pareto et des méthodes non agrégées non Pareto. Ce classement adopte un point de vue plus théorique articulé autour des notions d'agrégation et d'optimum de Pareto.

### II.3.1. Les méthodes agrégées

#### II.3.1.1. La moyenne pondérée

Cette méthode consiste à additionner tous les objectifs en affectant à chacun d'eux un coefficient de poids. Ce coefficient représente l'importance relative que le décideur attribue à l'objectif. Cela modifie un problème multi-objectif en un problème simple objectif de la forme :

$$\min \sum_{i=1}^k w_i f_i(x) \text{ avec } w_i \geq 0$$

$w_i$  Représente le poids affecté au critère  $i$  et  $\sum_{i=1}^k w_i = 1$

#### II.3.1.2. Méthode du but programmé

Cette méthode est également appelée *target vector optimisation*. Le décideur fixe un but  $T_i$  à atteindre pour chaque objectif  $f_i$ . Ces valeurs sont ensuite ajoutées au problème comme des contraintes supplémentaires. La nouvelle fonction objective est modifiée de façon à minimiser la somme des écarts entre les résultats et les buts à atteindre :

$$\min \sum_{i=1}^k |f_i(x) - T_i| \text{ avec } x \in F$$

$T_i$  représente la valeur à atteindre pour le  $i^{\text{ème}}$  objectif.

#### II.3.1.3. Méthode du but à atteindre

Dans cette approche le décideur spécifie l'ensemble des buts  $T_i$  qu'il souhaite atteindre et les poids associés  $w_i$ . La solution optimale est trouvée en résolvant le problème suivant :

$$\text{Minimiser } \alpha \text{ tel que } T_i + \alpha w_i \geq f_i(x) \text{ avec } \sum_{i=1}^k w_i = 1$$

Les objectifs  $T_i$  représentent le point de départ de la recherche dans l'espace et les poids  $w_i$  indiquent la direction de recherche dans l'espace.

#### II.3.1.4. Le min-max

Cette méthode est assez proche de la précédente. Elle minimise le maximum de l'écart relatif entre un objectif et son but associé par le décideur.

$$\min \max_i \left( \frac{f_i(x) - T_i}{T_i} \right) \text{ avec } i = 1, \dots, k$$

$T_i$  représente la valeur à atteindre pour le  $i^{\text{ème}}$  objectif.

#### II.3.1.5. La méthode $\varepsilon$ -contrainte

Cette méthode est basée sur la minimisation d'un objectif  $f_i$  en considérant que les autres objectifs  $f_j$  avec  $j \neq i$  doivent être inférieurs à une valeur  $\varepsilon_j$ . En général, l'objectif choisi est celui que le décideur souhaite optimiser en priorité.

$$\text{Minimiser } f_i(x) \text{ avec } f_j(x) \leq \varepsilon_j, \forall j \neq i$$

De cette manière, un problème simple objectif sous contraintes peut être résolu. Le décideur peut ensuite réitérer ce processus sur un objectif différent jusqu'à ce qu'il trouve une solution satisfaisante.

### II.3.2. Les méthodes Pareto :

Les approches Pareto utilisent directement la notion de dominance au sens de Pareto. Cette idée a été introduite initialement dans les algorithmes génétiques par Goldberg pour résoudre les problèmes proposés par Schaffer. Ce concept permet de ne pas favoriser un compromis plutôt qu'un autre mais il permet de fournir une aide précieuse au décideur. L'utilisation de la dominance est principalement à la base du développement des algorithmes génétiques multi-objectif présentés dans (II.4).

### II.3.3. les méthodes non agrégées, non Pareto

En général, les méthodes dites non agrégées et non Pareto possèdent un processus de recherche qui traite séparément les objectifs.

#### II.3.3.1. Sélection parallèle (VEGA)

En 1985 Schaffer propose une extension d'un algorithme génétique simple pour la résolution d'un problème multi-objectif [22]. Cette méthode est appelée *Vector Evaluated Genetic Algorithm*. La seule différence avec un algorithme génétique simple est la manière dont s'effectue la sélection. L'idée est simple. Si nous avons  $k$  objectifs et une population de  $n$  individus, une sélection de  $n/k$  individus est effectuée pour chaque objectif. Ainsi  $k$  sous-populations vont être créées, chacune d'entre elles contenant les  $n/k$  meilleurs individus pour un objectif particulier. Les  $k$  sous-populations sont ensuite mélangées afin d'obtenir une nouvelle population de taille  $n$ . Le processus se termine par l'application des opérateurs génétiques de modification (croisement et mutation).

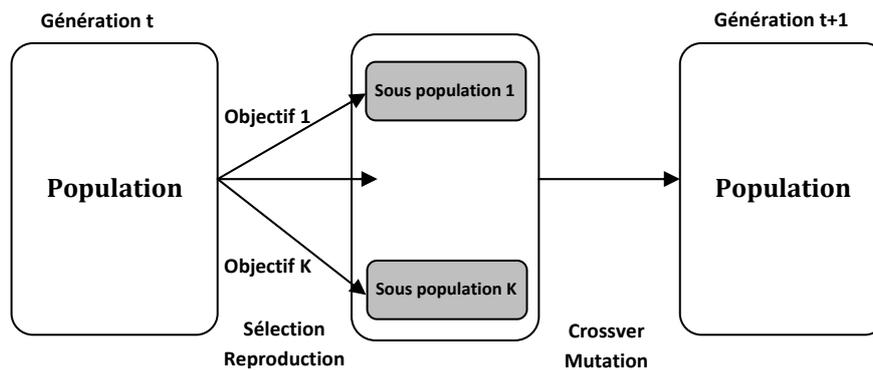


Figure II- 3 : Sélection parallèle dans l'algorithme VEGA

La méthode VEGA a tendance à créer des sous-populations dont les meilleurs individus sont spécialisés pour un objectif particulier. L'évolution de la population favorise l'apparition des espèces. En effet, comme la méthode de sélection ne tient compte que d'un seul objectif, elle privilégie les individus qui obtiennent une bonne performance pour cet objectif. Dès lors ces individus ne seront sélectionnés que lorsqu'on effectuera la sélection sur cet objectif. Les individus que Schaffer appelle les individus "milieu", parce qu'ayant une performance générale acceptable mais ne possédant aucun critère fort, vont être éliminés car ils ne seront sélectionnés dans aucune sous-population. Cette disparition entraîne la spécialisation des individus pour chaque objectif. Ce résultat est contraire au but initial de la méthode qui était de trouver un compromis entre les différents critères.

### II.3.3.2. Utilisation des genres

En 1992 Allenson propose une méthode qui utilise la notion de genre et d'attracteur sexuel pour traiter un problème à deux objectifs [23]. Son exemple d'application consiste à minimiser la longueur d'un pipeline tout en réduisant l'impact écologique de sa construction. En affectant un objectif à chaque genre, l'auteur espère minimiser les deux objectifs simultanément car un genre sera toujours jugé d'après l'objectif qui lui a été associé.

### II.3.3.3. La méthode lexicographique

Cette méthode, proposée par Fourman [24], classe les objectifs en fonction d'un ordre d'importance proposé par le décideur. Ensuite les fonctions objectif sont traitées dans cet ordre pour obtenir l'optimum.

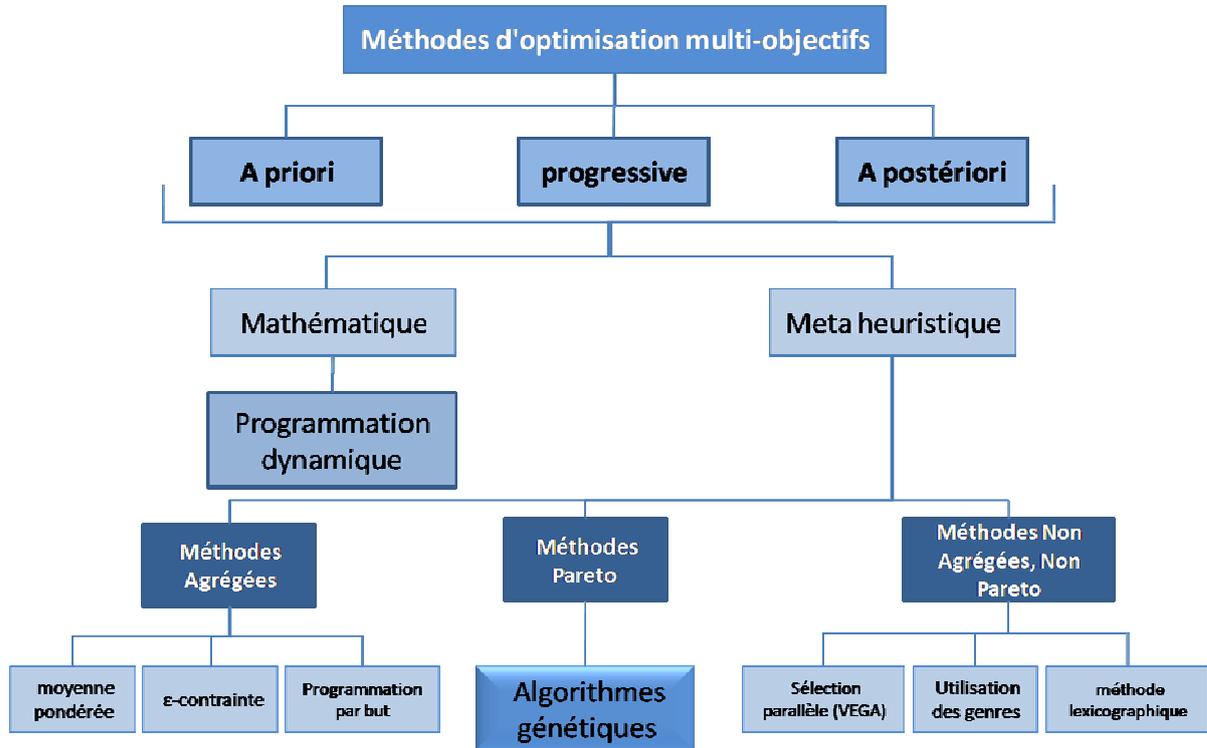


Figure II- 4 : Classification des méthodes d'optimisation multi-objectifs

## II.4. Les algorithmes génétiques multi-objectifs

### II.4.1. Principes généraux

Les algorithmes génétiques sont des algorithmes d'optimisation s'appuyant sur des techniques dérivées de la génétique et de l'évolution naturelle : croisements, mutations, sélection, etc. Les algorithmes génétiques ont déjà une histoire relativement ancienne, puisque les premiers travaux de John Holland sur les systèmes adaptatifs remontent à 1962 [38]. L'ouvrage de David Goldberg [39] a largement contribué à les vulgariser. Un algorithme génétique recherche le ou les extrema d'une fonction définie sur un espace de données. Pour l'utiliser, on doit disposer des cinq éléments suivants :

1. Un principe de codage de l'élément de population. Cette étape associe à chacun des points de l'espace d'état une structure de données. Elle se place généralement après une phase de modélisation mathématique du problème traité. Le choix du codage des données conditionne le succès des algorithmes génétiques. Les codages binaires ont été très employés à l'origine. Les codages réels sont désormais largement utilisés, notamment dans les domaines applicatifs, pour l'optimisation de problèmes à variables continues.

2. Un mécanisme de génération de la population initiale. Ce mécanisme doit être capable de produire une population d'individus non homogène qui servira de base pour les générations futures. Le choix de la population initiale est important car il peut rendre plus ou moins rapide la convergence vers l'optimum global. Dans le cas où l'on ne connaît rien du problème à résoudre, il est essentiel que la population initiale soit répartie sur tout le domaine de recherche.

3. Une fonction à optimiser. Celle-ci prend ses valeurs dans  $\mathbb{R}^+$  et est appelée fitness ou fonction d'évaluation de l'individu. Celle-ci est utilisée pour sélectionner et reproduire les meilleurs individus de la population.

4. Des opérateurs permettant de diversifier la population au cours des générations et d'explorer l'espace d'état. L'opérateur de croisement recompose les gènes d'individus existant dans la population, l'opérateur de mutation a pour but de garantir l'exploration de l'espace d'état.

5. Des paramètres de dimensionnement : taille de la population, nombre total de générations ou critère d'arrêt, probabilités d'application des opérateurs de croisement et de mutation.

Le principe général du fonctionnement d'un algorithme génétique est représenté sur la figure II-5.

On commence par engendrer une population d'individus de façon aléatoire. Pour passer d'une génération  $k$  à la génération  $k + 1$ , les trois opérations suivantes sont répétées pour tous les éléments de la population  $k$ . Des couples de parents  $P_1$  et  $P_2$  sont sélectionnés en fonction de leurs adaptations. L'opérateur de croisement leur est appliqué avec une probabilité  $P_c$  (généralement autour de 0.6) et engendre des couples d'enfants  $C_1$  et  $C_2$ . D'autres éléments  $P$  sont sélectionnés en fonction de leur adaptation. L'opérateur de mutation leur est appliqué avec la probabilité  $P_m$  ( $P_m$  est généralement très inférieur à  $P_c$ ) et engendre des individus mutés  $P'$ . Les enfants ( $C_1, C_2$ ) et les individus mutés  $P'$  sont ensuite évalués avant insertion dans la nouvelle population (la figure II-5 présente le cas où les enfants et les individus mutés remplacent les parents). Différents critères d'arrêt de l'algorithme peuvent être choisis :

– Le nombre de générations que l'on souhaite exécuter peut être fixé a priori. C'est ce que l'on est tenté de faire lorsque l'on doit trouver une solution dans un temps limité.

– L'algorithme peut être arrêté lorsque la population n'évolue plus ou plus suffisamment rapidement.

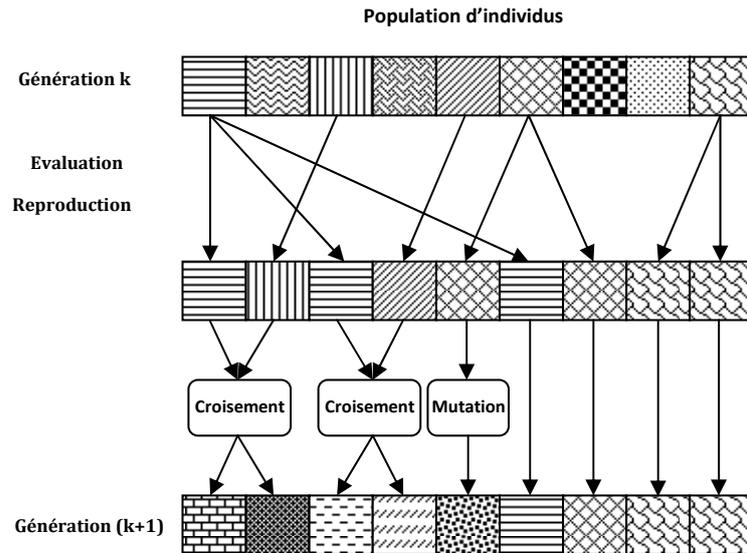


Figure II- 5 : Principe général des algorithmes génétiques

#### II.4.2. L'état de l'art de l'algorithme génétique multi-objectif

L'application des AG à l'optimisation multi-objectif est récente. D'après Veldhuizen & Lamont dans *Analyzing the State-of-the Art*, publié en 2000, le premier travail a été proposé par Schaffer en 1984. Dès lors, le nombre de publications dans ce domaine grandit exponentiellement [19]. En 1999, Coello a présenté une étude classifiant et évaluant diverses techniques d'optimisation multi-objectif. Cette étude a conclu que la plupart des méthodes sont basées sur les AG monocritère. Les différences portent sur la méthode de sélection des individus. D'après plusieurs auteurs déjà mentionnés, les procédures principales sont, dans l'ordre chronologique de publication :

- L'AG basé sur un vecteur d'évaluation (Vector Evaluated Genetic Algorithm - VEGA) 1984 - Schaffer, Schaffer : on modifie l'opérateur génétique de sélection d'un AG monocritère de façon à créer des populations séparées pour chaque objectif. Ceci génère des « spécialisations », c'est-à-dire que chaque population tendra vers le point optimal de chaque objectif et non vers la frontière Pareto-optimale associée au problème ;
- L'AG Multi-objectif (Multi-objective Genetic Algorithm - MOGA), 1993- Fonseca & Fleming [26]. L'idée du MOGA est d'établir un ordre des individus: les non-dominés ont des classifications semblables, tandis les dominés sont pénalisés d'après leur dominance. La difficulté est de trouver une façon d'interpoler ces deux groupes de manière à permettre une bonne conformation de la frontière Pareto-optimale ;
- L'AG basé sur le tri des non-dominés (Nondominated Sorting Genetic Algorithm) NSGA en 1993 - Srinivas & Deb, NSGA II en 2000 - Deb et al..

Dans ce cas, seules les solutions non-dominées sont sélectionnées. Comme ces solutions sont efficaces, elles auront la même probabilité de se reproduire. Dans cette méthodologie, l'inconvénient est la difficulté de maintenir la diversité de la population. L'absence de diversité peut générer une frontière incomplète, c'est-à-dire une concentration des solutions dans certaines régions ;

- L'AG basé sur le nichage de Pareto (Niche Pareto Genetic Algorithm-NPGA) 1993 - Horn & Nafpliotis [27]. L'une des techniques de sélection pour l'AG monocritère est le tournoi entre les individus. Horn & Nafpliotis ont implémenté un tournoi dans lequel la règle de compétition est liée à la dominance de Pareto. Dans cette méthode, il est difficile de déterminer combien d'individus participeront du tournoi, et lesquels ;
- La méthode des populations intermédiaires, 1996 - Viennet & Marc [28]. Cet algorithme a trois étapes : la détermination des points minimums pour chaque objectif ; la recherche de la population intermédiaire (basée sur le choix de  $n$  individus pour chaque objectif) ; et à partir de cette population, la définition de la frontière Pareto-optimale. La principale restriction de cette méthode est que chaque objectif doit avoir un optimum unique dans l'espace d'étude ;
- L'Algorithme évolutionnaire basé sur la « force » de Pareto (Strength Pareto Evolutionary Algorithm) - SPEA I 1998 - Zitzler & Thiele [29], SPEA II 1999 et 2001- Zitzler [30], Zitzler et al. [31]. Un numéro indiquant une mesure de «force» pour chaque individu est créé par le processus de sélection. Les individus non-dominés doivent posséder une « force » plus grande.

Il existe déjà plusieurs travaux confrontant les diverses méthodologies d'AGMO. Parmi les plus récents, on peut citer : Zitzler en 1999, Dias en 2000, Dias & Vasconcelos en 2002 et Régnier en 2003. D'après ce dernier, les méthodes SPEA et NSGA méritent une attention particulière : «le SPEA-II et le NSGA-II semblent être aujourd'hui des techniques de référence solides... ».

D'autres travaux ont été développés de façon à compléter ces méthodologies. On peut par exemple citer l'incorporation de contraintes à l'espace d'étude (proposé en 2003 par Vieira et al. ou la possibilité de résoudre en calcul parallèle (présenté en 2003 par Veldhuizen et al.). Plusieurs méthodologies de test de performance ont également été proposées, comme celles publiées par Veldhuizen & Lamont (2000), ou par Takahashi et al. (2003).

### II.4.3. Description détaillée

#### II.4.3.1. Codage des données

Historiquement, le codage utilisé par les algorithmes génétiques était représenté sous forme de chaînes de bits contenant toute l'information nécessaire à la description d'un point dans l'espace d'état. Ce type de codage a pour intérêt de permettre de créer des opérateurs de croisement et de mutation simples. C'est également en utilisant ce type de codage que les premiers résultats de convergence théorique ont été obtenus.

Cependant, ce type de codage n'est pas toujours bon :

- deux éléments voisins en terme de distance de Hamming ne codent pas nécessairement deux éléments proches dans l'espace de recherche. Cet inconvénient peut être évité en utilisant un codage de Gray.

- Pour des problèmes d'optimisation dans des espaces de grande dimension, le codage binaire peut rapidement devenir mauvais. Généralement, chaque variable est représentée par une partie de la chaîne de bits et la structure du problème n'est pas bien reflétée, l'ordre des variables ayant une importance dans la structure du chromosome, alors qu'il n'en a pas forcément dans la structure du problème.

Les algorithmes génétiques utilisant des vecteurs réels évitent ce problème en conservant les variables du problème dans le codage de l'élément de population, sans passer par le codage binaire intermédiaire. Certains les appellent RCGA (Real Coded Genetic Algorithms), d'autres parlent d'algorithmes évolutionnaires. La structure du problème est conservée dans le codage.

#### II.4.3.2. Génération aléatoire de la population initiale

Le choix de la population initiale d'individus conditionne fortement la rapidité de l'algorithme. Si la position de l'optimum dans l'espace d'état est totalement inconnue, il est naturel d'engendrer aléatoirement des individus en faisant des tirages uniformes dans chacun des domaines associés aux composantes de l'espace d'état, en veillant à ce que les individus produits respectent les contraintes. Si par contre, des informations a priori sur le problème sont disponibles, il paraît bien évidemment naturel d'engendrer les individus dans un sous-domaine particulier afin d'accélérer la convergence. Dans l'hypothèse où la gestion des contraintes ne peut se faire directement, les contraintes sont généralement incluses dans le critère à optimiser sous forme de pénalités.

#### II.4.3.3. Gestion des contraintes

Un élément de population qui viole une contrainte se verra attribuer une mauvaise fitness et aura une probabilité forte d'être éliminé par le processus de sélection. Il peut cependant être intéressant de conserver, tout en les pénalisant, les éléments non admissibles car ils peuvent permettre de générer des éléments admissibles de bonne qualité. Pour de nombreux problèmes, l'optimum est atteint lorsque l'une au moins des contraintes de séparation est saturée, c'est-à-dire sur la frontière de l'espace admissible.

Gérer les contraintes en pénalisant la fonction fitness est difficile, un “dosage” s’impose pour ne pas favoriser la recherche de solutions admissibles au détriment de la recherche de l’optimum ou inversement. Disposant d’une population d’individus non homogène, la diversité de la population doit être entretenue au cours des générations, afin de parcourir le plus largement possible l’espace d’état. C’est le rôle des opérateurs de croisement et de mutation.

#### II.4.3.4. Opérateur de croisement

Le croisement a pour but d’enrichir la diversité de la population en manipulant la structure des chromosomes. Classiquement, les croisements sont envisagés avec deux parents et génèrent deux enfants. Initialement, le croisement associé au codage par chaînes de bits est le croisement à découpage de chromosomes (slicing crossover). Pour effectuer ce type de croisement sur des chromosomes constitués de  $M$  gènes, on tire aléatoirement une position dans chacun des parents. On échange ensuite les deux sous-chaînes terminales de chacun des deux chromosomes, ce qui produit deux enfants  $C_1$  et  $C_2$  (voir figure II-5). On peut étendre ce principe en découpant le chromosome non pas en 2 sous chaînes mais en 3, 4, etc. (voir figure II-6).

Ce type de croisement à découpage de chromosomes est très efficace pour les problèmes discrets. Pour les problèmes continus, un croisement “barycentrique” est souvent utilisé : deux gènes  $P_1(i)$  et  $P_2(i)$  sont sélectionnés dans chacun des parents à la même position  $i$ . Ils définissent deux nouveaux gènes  $C_1(i)$  et  $C_2(i)$  par combinaison linéaire :

$$\begin{cases} C_1(i) = \alpha P_1(i) + (1 - \alpha)P_2(i) \\ C_2(i) = (1 - \alpha)P_1(i) + \alpha P_2(i) \end{cases}$$

où  $\alpha$  est un coefficient de pondération aléatoire adapté au domaine d’extension des gènes (il n’est pas nécessairement compris entre 0 et 1, il peut par exemple prendre des valeurs dans l’intervalle  $[-0.5, 1.5]$ , ce qui permet d’engendrer des points entre, ou à l’extérieur des deux gènes considérés).

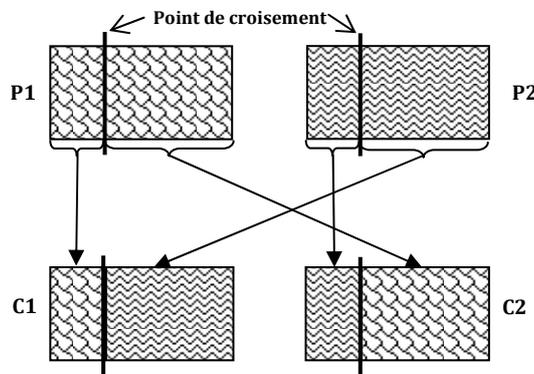


Figure II- 6 : Croisement à 1 point

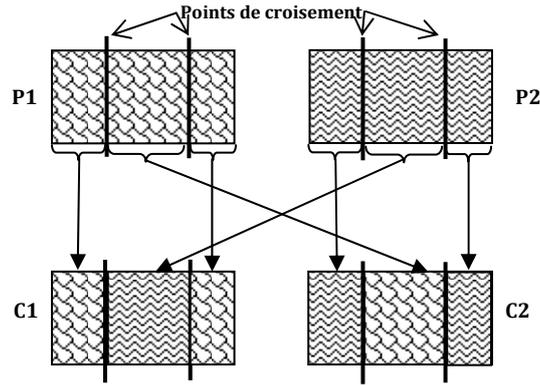


Figure II- 7 : Croisement à 2 points

Dans le cas particulier d'un chromosome matriciel constitué par la concaténation de vecteurs, on peut étendre ce principe de croisement aux vecteurs constituant les gènes (voir figure II-7) :

$$\begin{cases} \vec{C}_1(i) = \alpha \vec{P}_1(i) + (1 - \alpha) \vec{P}_2(i) \\ \vec{C}_2(i) = (1 - \alpha) \vec{P}_1(i) + \alpha \vec{P}_2(i) \end{cases}$$

On peut imaginer et tester des opérateurs de croisement plus ou moins complexes sur un problème donné mais l'efficacité de ces derniers est souvent liée intrinsèquement au problème.

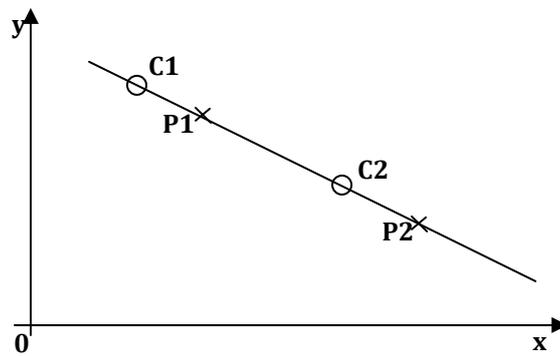


Figure II- 8 : Croisement barycentrique

#### II.4.3.5. Opérateur de mutation

L'opérateur de mutation apporte aux algorithmes génétiques la propriété d'ergodicité de parcours d'espace. Cette propriété indique que l'algorithme génétique sera susceptible d'atteindre tous les points de l'espace d'état, sans pour autant les parcourir tous dans le processus de résolution. Ainsi en toute rigueur, l'algorithme génétique peut converger sans croisement, et certaines implémentations fonctionnent de cette manière. Les propriétés de convergence des algorithmes génétiques sont donc fortement dépendantes de cet opérateur sur le plan théorique. Pour les problèmes discrets, l'opérateur de mutation consiste généralement à tirer aléatoirement un gène dans le chromosome et à le remplacer par une valeur aléatoire (voir figure II-8). Si la notion de distance existe, cette valeur peut être choisie dans le voisinage de la valeur initiale.

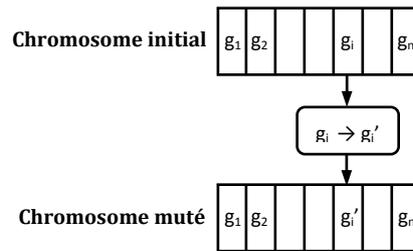


Figure II- 9 : Principe de l'opérateur de mutation

#### II.4.3.6. Principes de sélection

A l'inverse d'autres techniques d'optimisation, les algorithmes génétiques ne requièrent pas d'hypothèse particulière sur la régularité de la fonction objective. L'algorithme génétique n'utilise notamment pas ses dérivées successives, ce qui rend très vaste son domaine d'application. Aucune hypothèse sur la continuité n'est non plus requise. Néanmoins, dans la pratique, les algorithmes génétiques sont sensibles à la régularité des fonctions qu'ils optimisent. Le peu d'hypothèses requises permet de traiter des problèmes très complexes. La fonction à optimiser peut ainsi être le résultat d'une simulation. La sélection permet d'identifier statistiquement les meilleurs individus d'une population et d'éliminer les mauvais. On trouve dans la littérature un nombre important de principes de sélection plus ou moins adaptés aux problèmes qu'ils traitent. Les deux principes de sélection suivants sont les plus couramment utilisés : *Roulette wheel selection* et *Stochastic remainder without replacement selection*.

Le principe de *Roulette wheel selection* consiste à associer à chaque individu un segment dont la longueur est proportionnelle à sa fitness. On reproduit ici le principe de tirage aléatoire utilisé dans les roulettes de casinos avec une structure linéaire. Ces segments sont ensuite concaténés sur un axe que l'on normalise entre 0 et 1. On tire alors un nombre aléatoire de distribution uniforme entre 0 et 1, puis on "regarde" quel est le segment sélectionné. Avec ce système, les grands segments, c'est-à-dire les bons individus, seront plus souvent choisis que les petits. Lorsque la dimension de la population est réduite, il est difficile d'obtenir en pratique l'espérance mathématique de sélection en raison du peu de tirages effectués. Un biais de sélection plus ou moins fort existe suivant la dimension de la population. La *Stochastic remainder without replacement selection* évite ce genre de problème et donne de bons résultats pour nos applications. Décrivons ce principe de sélection :

- Pour chaque élément  $i$ , on calcule le rapport  $r_i$  de sa fitness sur la moyenne des fitness.
- Soit  $e(r_i)$  la partie entière de  $r_i$ , chaque élément est reproduit exactement  $e(r_i)$  fois.
- La *Roulette wheel selection* précédemment décrite est appliquée sur les individus affectés des fitness  $r_i - e(r_i)$ .

Lorsque des populations de faible taille sont utilisées, ce principe de sélection s'avère généralement efficace dans les applications pratiques.

### II.5. Présentation de l'algorithme NSGA II

### II.5.1. Non dominated Sorting Genetic Algorithm (NSGA)

Dans la méthode proposée par Srivinas et Deb en 1993 [21], le calcul de la fitness s'effectue en séparant la population en plusieurs groupes en fonction du degré de domination au sens de Pareto de chaque individu.

- **Algorithme de la fonction de notation**

- a) Dans la population entière, on recherche les individus non dominés. Ces derniers constituent la première frontière de Pareto.
- b) On leur attribue une valeur de fitness factice. Cette valeur est supposée donner une chance égale de reproduction à tous ces individus. Mais pour maintenir la diversité dans la population, il est nécessaire d'appliquer une fonction de sharing sur cette valeur.
- c) Ensuite, ce premier groupe d'individus est supprimé de la population.
- d) On recommence cette procédure pour déterminer la seconde frontière de Pareto. La valeur factice de fitness attribuée à ce second groupe est inférieure à la plus petite fitness après application de la fonction de sharing sur le premier groupe. Ce mécanisme est répété jusqu'à ce que l'on ait traité tous les individus de la population.

L'algorithme se déroule ensuite comme un algorithme génétique classique. Srivinas et Deb utilisent une sélection basée sur le reste stochastique. Mais leur méthode peut être utilisée avec d'autres heuristiques de sélections (tournoi, roulette pipée, etc.).

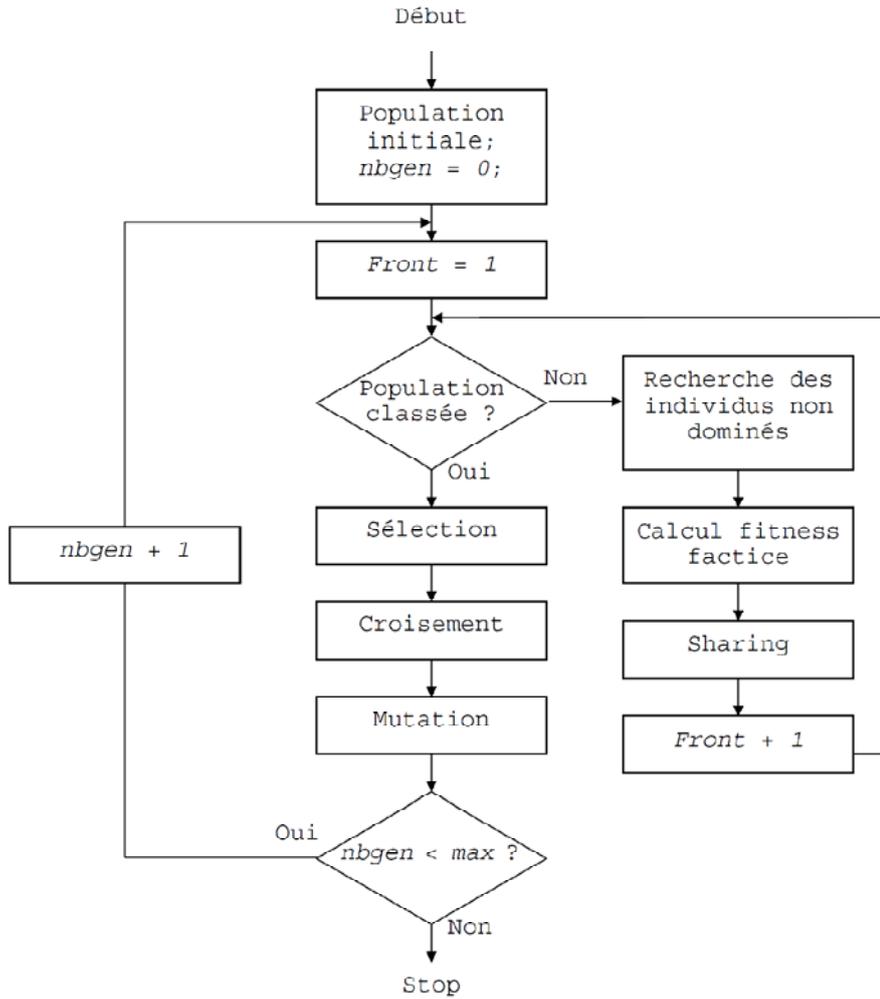


Figure II- 10 : Schéma de fonctionnement de NSGA

- **Critique**

Cette méthode paraît moins efficace en temps de calcul que la méthode MOGA car le temps de calcul de la notation (tri de la population et sharing) est important. Mais l'utilisation d'un sharing sur l'espace d'état et le tri des solutions en différentes frontières semblent plus appropriés à maintenir une grande diversité de la population et à répartir plus efficacement les solutions sur la frontière de Pareto. De plus, cette méthode est applicable dans des problèmes avec un nombre quelconque d'objectifs.

### II.5.2. NSGA II

« NSGA-II » a été proposé par Deb et al. [32] pour répondre aux critiques qui avaient été faites à NSGA : complexité, non élitisme et utilisation du sharing.

La complexité de l'algorithme NSGA est notamment due à la procédure de création des différentes frontières. Pour diminuer la complexité de calcul de NSGA, Deb propose une modification de la procédure de tri de la population en plusieurs frontières.

- Algorithme

- Pour chaque solution il calcule deux paramètres :
  - $n_i$ , le nombre de solutions qui dominent la solution  $i$
  - $S_i$ , l'ensemble des solutions que la solution  $i$  domine.
- Il identifie les points tels que  $n_i = 0$ . Ces points forment la frontière  $F_1$ .
- Pour chaque élément  $i$  de  $F_1$ , il visite l'ensemble  $S_i$ , et il retranche 1 au  $n_j$  de chaque élément  $j$  de  $S_i$ .
- Il recommence en c) jusqu'à ce que tous les points soient traités.

L'autre critique sur NSGA est l'utilisation du sharing, méthode qui exige le réglage d'un ou plusieurs paramètre(s) et qui est également forte consommatrice de calculs. Dans NSGA II, Kalyanmoy Deb remplace la fonction de sharing par une fonction de crowding. Il attribue deux caractéristiques à chaque individu :

- $i_{rank}$  représente le rang de non domination de l'individu. Cette caractéristique dépend de la frontière à laquelle appartient l'individu.
- $i_{distance}$  représente la distance de crowding de l'individu et permet d'estimer la densité de la population autour de lui.

Pour estimer la densité au voisinage d'une solution  $i$ , il calcule la distance moyenne sur chaque objectif, entre les deux points les plus proches situés de part et d'autre de la solution. Cette quantité appelée  $i_{distance}$  sert d'estimateur de taille du plus large hypercube incluant le point  $i$  sans inclure un autre point de la population. Cette distance de crowding va être utilisée pour guider le processus de sélection.

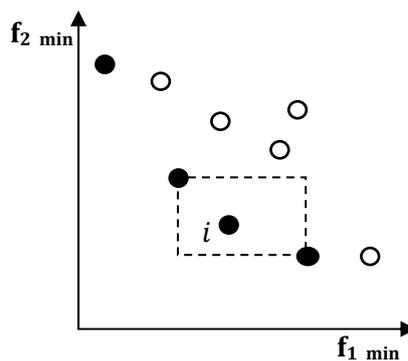


Figure II- 11 : Calcul de la distance de crowding.

Pour répondre à la critique de non élitisme, Deb utilise dans cette méthode une sélection par tournoi et modifie la procédure de passage entre deux générations. Si deux solutions sont sélectionnées pour participer au tournoi, la solution de plus bas rang  $i_{rank}$  sera retenue. Mais si les deux rangs sont identiques, il est préférable d'utiliser le point

situé dans une région dépeuplée, c'est-à-dire avec une valeur  $i_{distance}$  importante. Deb définit sa notion de préférence entre deux solutions de la façon suivante :

$$i \geq j \text{ si } (i_{rank} < j_{rank}) \text{ ou } ((i_{rank} = j_{rank}) \text{ et } (i_{distance} > j_{distance}))$$

NSGA-II fonctionne de la manière suivante : sa population  $R_t$ , où  $t$  est le numéro de la génération courante, est divisée en deux sous-populations  $P_t$  et  $Q_t$ . Les tailles de  $P_t$  et  $Q_t$  sont égales à  $N$  et la taille de  $R_t$  est donc  $2N$ . La sous-population  $P_t$  correspond aux parents et  $Q_t$  aux enfants après la phase de recombinaison. Nous allons maintenant présenter les principales étapes d'une génération de NSGA II. Les principales étapes d'une génération  $t$  de l'algorithme sont [40]:

- **Etape 1 :** Combiner les populations de parents et d'enfants et créer  $R_t = P_t \cup Q_t$ . Appliquer un tri non-dominé sur  $R_t$  et identifier les différents fronts  $\mathcal{F}_i$ ,  $i = 1, \dots, etc.$
- **Etape 2 :** Créer une nouvelle population  $P_{t+1} = \emptyset$ . Initialiser  $i = 1$ . Tant que  $|P_{t+1}| + |\mathcal{F}_i| < N$ , on fait  $P_{t+1} = P_{t+1} + \mathcal{F}_i$  et  $i = i + 1$ .
- **Etape 3 :** Lorsque toutes les solutions de  $\mathcal{F}_i$  ne peuvent être intégrées à  $P_{t+1}$ , on fait un tri sur les distances de crowding et on inclut à  $P_{t+1}$  les  $N - |P_{t+1}|$  solutions les plus éparpillées en utilisant la distance de crowding dans l'ensemble  $\mathcal{F}_i$ .
- **Etape 4 :** Créer la population d'enfants  $Q_{t+1}$  à partir de  $P_{t+1}$  en utilisant la sélection par tournoi à base de crowding et les opérateurs génétiques.

Le déroulement d'une génération est résumé dans la Figure II-12.

Dans l'étape 1, le tri non-dominé utilise la notion de profondeur de dominance. Les distances de crowding de l'étape 2 permettent d'améliorer la diversité de l'ensemble généré durant la recherche. Cette mesure donne l'estimation de la densité de solutions entourant une solution  $i$  dans la population. Cette densité est donnée par une approximation du périmètre du cuboïde formé par les voisins appartenant au même ensemble  $\mathcal{F}_i$  les plus proches de  $i$  (Figure II-11). La distance de crowding des solutions d'un ensemble  $\mathcal{F}_i$  donné se calcule par l'algorithme II-1.

La sélection par tournoi de l'étape 3 fonctionne de la manière suivante. Un individu  $i$  possède deux attributs : son rang  $r_i$  donné par le tri non-dominé et une distance de crowding  $d_i$ . Une solution  $i$  gagne le tournoi contre un individu  $j$  si l'une des conditions suivantes est vraie :

1.  $r_i < r_j$
2.  $r_i = r_j$  et  $d_i > d_j$ .

La première condition assure que la solution choisie appartient à un meilleur ensemble non-dominé. En cas d'égalité, la seconde condition permet de choisir la solution qui apporte le plus de diversification.

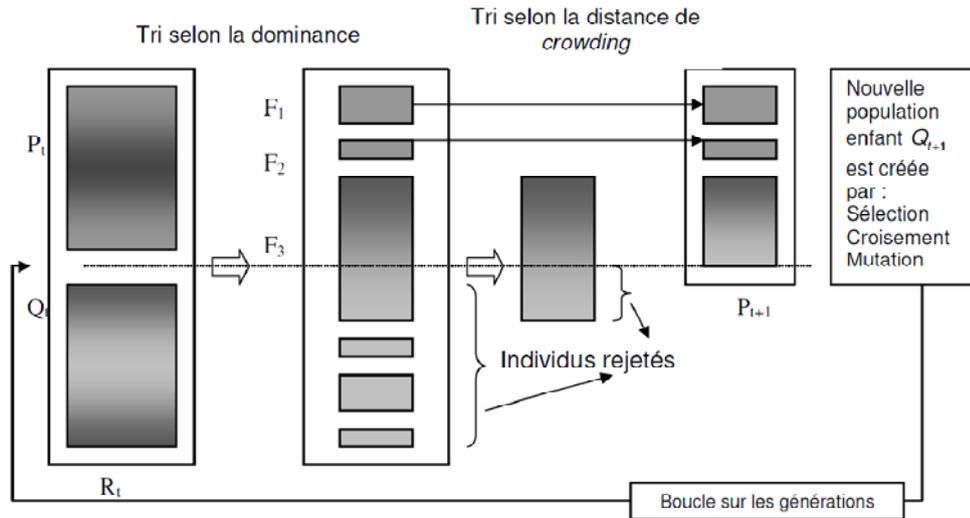


Figure II- 12 : L'algorithme évolutionnaire NSGA II [40]

**Algorithme de NSGA II**

1. La première génération  $P_0$  de taille  $N$  est créée aléatoirement.
2. Sélection par tournoi en utilisant la préférence définie ci-dessus et application des opérateurs de modification pour créer un ensemble d'enfants  $Q_t$  de taille  $N$ .
3. On mélange  $P_t$  et  $Q_t$  :  $R_t = P_t \cup Q_t$ .
4. Calcul de toutes les frontières  $F_i$  de  $R_t$  et ajout dans  $P_{t+1}$  jusqu'à ce que la taille de  $P_{t+1}$  soit égale à  $N$ .

Les éléments de la dernière frontière calculée sont triés en fonction de la préférence citée ci-dessus. Seuls les éléments permettant à  $P_{t+1}$  d'atteindre la taille  $N$  sont sélectionnés.

5. On recommence en 2.

**Algorithme II-1 : Calcul de la distance de crowding.**
 $l = |\mathcal{F}_i|$ 

*Pour tout objectif  $m$  faire*

Trier  $\mathcal{F}_i$  par rapport aux valeurs de l'objectif  $f_m$  par ordre décroissant. On obtient alors le vecteur d'indices triés  $I^m$ .

*Fin pour*

 $d_{I_1^m} = d_{I_l^m} = \infty$ 

*Pour  $j = 2, \dots, l - 1$  faire*

$$d_{I_j^m} = \frac{f_m(s_{I_{j+1}^m}) - f_m(s_{I_{j-1}^m})}{f_m^{\max} - f_m^{\min}}$$

*Fin pour*

La solution de NSGA II est l'ensemble des solutions non-dominées de la population finale. Cependant, si le nombre de solutions Pareto optimales est plus grand que  $N$ , certaines de ces solutions seront perdues lors de la recherche. Pour palier à ce problème, une archive a été ajoutée et permet de conserver l'ensemble des solutions potentiellement

Pareto optimales trouvées au cours de l'exécution de l'algorithme. Cette archive n'est utilisée que dans un but de stockage et les solutions qui s'y trouvent ne sont jamais utilisées durant le déroulement de l'algorithme. C'est notamment le cas lorsque l'on ajoute la diversification élitiste à NSGA II comme cela est présenté ci-dessous.

## II.6. Conclusion

Dans ce chapitre, nous avons essayé de décrire les problèmes d'optimisation multi-objectifs, tout en montrant la manière de définir un problème pareil, ses contraintes, ses objectifs ainsi que les méthodes utilisées pour le résoudre tout en respectant le concept de Compromis et les frontières de Pareto. Nous avons parlé sur les méthodes utilisées dans le domaine de l'optimisation multi-objectifs soit les problèmes mathématiques ou les méta-heuristiques, parmi les méta-heuristiques existant, nous trouvant les algorithmes génétiques qui exercent le principe de la génétique pour résoudre ces problèmes et ils ont démontrés leur efficacité.

Pour cette raison et puisque notre travail se base sur l'un des algorithmes génétiques nous avons présenté un état de l'art sur les algorithmes génétiques d'une manière générale, ensuite nous avons présenté le fonctionnement du algorithme génétique et comment l'utilise pour résoudre les problèmes d'optimisation multi-objectifs.

Enfin, nous avons présenté l'algorithme NSGA II qui est la base de la « *Genetic and Evolutionary Algorithm Toolbox (GEAtbx)* » que nous l'utiliserons dans notre travail.

# Chapitre III

## Modélisation du robot Delta

### III.1. introduction

La synthèse de mécanismes consiste à trouver un meilleur mécanisme au sens d'un ou de plusieurs critères prédéfinis, pour accomplir une tâche donnée. La première étape de la synthèse de mécanismes est de modéliser la tâche qui caractérise l'espace de recherche des solutions [33]. Pour un robot, cette tâche est l'ensemble des positions à atteindre pouvant être une liste de points de passage, une trajectoire à parcourir, une surface à laquelle le robot doit accéder et/ou un volume de travail. Les différentes étapes de notre démarche de conception sont :

- sélection d'une famille d'architectures parallèles correspondant à la tâche à effectuer (permet de trouver la meilleure topologie à l'intérieur de cette famille)
- Etude de l'évolution de plusieurs indicateurs en fonction d'un vecteur réduit de paramètres de la topologie choisie.
- Fixation à l'aide de ces indicateurs des dimensions du mécanisme qui agissent sur sa cinématique.
- Dimensionnement des organes qui le composent

### III.2. Architectures parallèles candidates pour les machines-outils.

#### III.2.1. Limitations de machines outil d'une série

Les machines outil récentes présentent des améliorations technologiques au niveau des axes, des actionneurs linéaires, des portées. La plupart de ces machines outil sont basées sur une architecture d'une série (Chiffre 5), dont les rapports simples de l'entrée/sortie sont des avantages très appréciés. Toutefois les masses lourdes à être porté et déplacé par les axes numériques limitent les performances dynamiques (vitesses d'entraînement, accélérations), ce qui a incité les fabricants des machines outil à opter pour les machines agiles de types parallèle dès les années 1990 [34].

#### III.2.2. Choix d'une architecture

De part la nature des performances des machines-outils et des robots, nous pouvons établir un rapprochement entre ces deux types de machines, en notons toutefois que la valeur de ces performances n'est pas du même ordre de grandeur. Par conséquent, cette différence ne permet qu'à certaines architectures de robots parallèles d'être adaptées pour les machines-outils, dont les performances dynamiques, la précision, le comportement thermique et la rigidité sont des points très importants.

**Les performances dynamiques** des machines-outils conditionnent leur productivité. L'amélioration des performances dynamiques passe obligatoirement par une réduction des masses en mouvement. Le principal intérêt d'utiliser des mécanismes parallèles à la place des architectures série réside dans le fait que leurs performances dynamiques sont intrinsèquement meilleures. Afin de profiter au maximum de l'amélioration de ces performances, nous devons choisir des architectures dont les pièces en mouvement sont les plus légères possibles, c'est à dire les structures dont les actionneurs (qui comptent parmi les pièces les plus massives) sont fixés sur le bâti de la machine.

**Comportement thermique :** La précision des machines est un autre point important qui est influencée par la qualité de réalisation de leurs organes, par la finesse de leur étalonnage ainsi que par leur comportement thermique. L'énergie thermique dissipée par les moteurs provoque une augmentation de la température des « pattes » de la machine, cette augmentation de température induit un allongement des barres qui nuit au positionnement précis de l'outil. La solution idéale est d'évacuer l'énergie thermique dissipée par les moteurs dans le bâti.

**La rigidité** des organes de la chaîne cinématique reliant la broche à l'outil est un point capital pour les machines-outils et ce afin que les efforts de coupe ne provoquent pas un déplacement de l'outil trop important entraînant un défaut de la géométrie et de la position de la surface usinée, même si ces efforts de coupe diminuent en usinage TGV. Pour obtenir une machine rigide, chacune des pièces chargées du mécanisme doit avoir une déformation minimale ce qui implique soit, l'utilisation de pièces massives au détriment de la dynamique de la machine, soit, un chargement des pièces favorable par rapport à leur géométrie. Nous voyons que, par rapport à notre problème, la deuxième solution est préférable.

Enfin, concernant la simplicité de réalisation, il est préférable d'utiliser des actionneurs fixes afin d'éviter la déformation des câbles qui les alimentent et éventuellement celle des canalisations qui les refroidissent.

Les points suivants ressortent de notre phase de réflexion sur la conception d'une machine-outil parallèle performante :

- Actionneurs fixes sur le bâti
- Utilisation d'actionneurs linéaires
- Chaînes cinématiques identiques

- Pas de chaîne passive
- Utilisation de barres de longueur constante.

Compte tenu de la réduction de l'espace de recherche à laquelle nous venons de procéder, il ne reste, parmi les architectures parallèles connues à ce jour, que deux architectures candidates : l'architecture Delta dont la nacelle possède trois degrés de liberté de translation et l'architecture P-U-S à six degrés de liberté, cette architecture ayant déjà été utilisée pour la réalisation de machines outils (HexaM) [35]. En fonction de la nature des tâches que devra effectuer la machine à construire (usinage trois axes) on va choisir l'architecture Delta, pour cette dernière, trouver le meilleur type d'agencement. Pour cela, nous modélisons de manière générique l'architecture choisie afin d'évaluer un ou plusieurs critères de performance.

### III.3. Modélisation générique

#### III.3.1. Paramétrage

##### Terminologie :

Nous appellerons « barres » les pièces reliant la nacelle aux actionneurs. Pour un agencement de type Delta, ces barres sont regroupées par paires.

Les mécanismes étudiés comportent une partie fixe reliée à une partie mobile (nacelle) par plusieurs barres. Nous utilisons deux repères

- Un repère  $\mathcal{R}_b$  fixe, attaché arbitrairement au centre du volume de travail, soit le point O.
- Un repère  $\mathcal{R}_n$  mobile, attaché à la nacelle. Nous désignerons par position de la nacelle l'origine du repère  $\mathcal{R}_n$  dont les coordonnées sont exprimées dans  $\mathcal{R}_b$ , soit le point E.
- Pour un agencement de type Delta, la disposition des articulations de la machine impose au repère  $\mathcal{R}_n$  de rester parallèle au repère  $\mathcal{R}_b$ . Pour simplifier notre exposé, les axes de  $\mathcal{R}_n$  seront choisis parallèles et de même sens que ceux de  $\mathcal{R}_b$ .

#### III.3.2. la description géométrique

Les barres d'une même paire reliant la nacelle aux actionneurs sont toutes deux de longueur  $l_i$ . Elles sont attachées d'un côté, à la nacelle de part et d'autre du point  $B_i$  (points  $B_{i1}$  et  $B_{i2}$ ) et de l'autre côté, à l'actionneur numéro  $i$  de part et d'autre du point  $A_i$  (points  $A_{i1}$  et  $A_{i2}$ ).

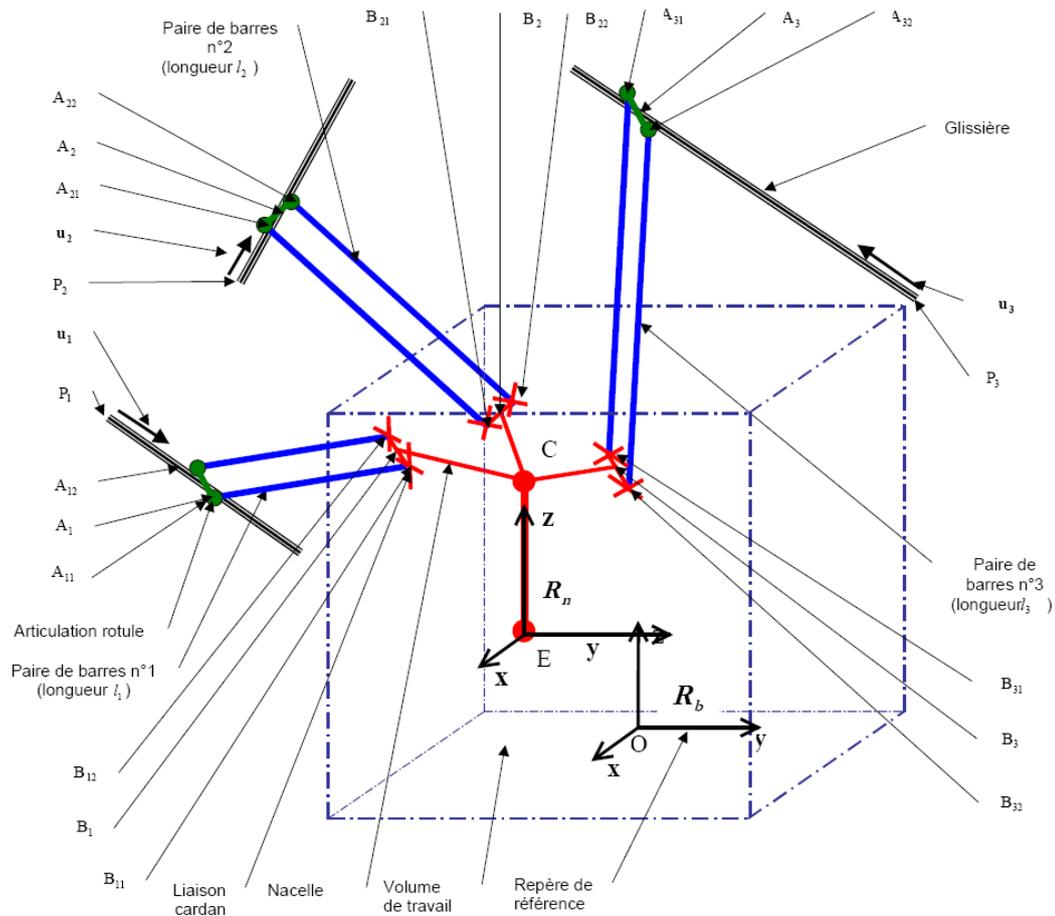


Figure III- 1 : Paramétrage de la géométrie de structures de type Delta linéaire [35]

La Figure III-2 décrit la position des différents points considérés sur la nacelle :

- Le point G est le centre de gravité (situé sur l'axe de symétrie) de l'ensemble nacelle + électro-broche.
- Le point C est le centre de la nacelle.
- Le point D est le point correspondant à la face avant du cône de la broche. Les vitesses de la nacelle sont mesurées en ce point.
- Le point E est le point piloté. La position de la nacelle est donnée par les coordonnées de ce point.
- Le point F est le point d'application des efforts de coupe.

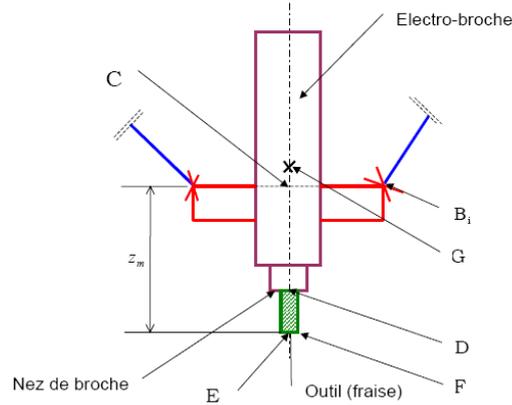


Figure III- 2 : Points de la nacelle.

### Hypothèses simplificatrices

Lors de toutes les études géométriques, cinématiques et dynamiques, les articulations seront considérées idéales (sans jeu, sans élasticité et sans frottement) et les corps seront supposés géométriquement exacts et parfaitement rigides.

### III.3.3. Modélisation géométrique

Les modèles géométriques direct et inverse sont les relations qui expriment la situation de l'organe terminal du robot (la nacelle dans le cas présent) en fonction de la configuration du mécanisme (coordonnées articulaires) et inversement. [36]

Dans notre cas de l'architecture Delta l'orientation reste constante (le paramétrage de l'orientation de la nacelle est inutile car cette dernière reste parallèle à un plan de référence) et seulement trois actionneurs sont utilisés. Nous devons alors résoudre un système de trois équations à trois inconnues.

Soit :

- $P$  la matrice contenant les coordonnées des points  $P_i$  exprimées dans le repère  $\mathcal{R}_b$
- $A$  la matrice contenant les coordonnées des points  $A_i$  exprimées dans le repère  $\mathcal{R}_b$ .
- $U$  la matrice contenant les composantes des vecteurs  $u_i$  exprimées dans le repère  $\mathcal{R}_b$ .

#### III.3.3.1. Modèle géométrique inverse

Pour calculer le vecteur des variables articulaires  $q$  (les coordonnées suivant  $\mathbf{z}$  des points A), nous écrivons que chacun des segments  $A_iB_i$ , pour  $1 \leq i \leq k$  ( $k=3$ ) a une longueur constante  $l_i$ . Ce qui nous donne le système de trois équations

$$q_i^2 - q_i[2(x - P_iB_i)u_i] + [(x - P_iB_i)^2 - l_i^2] = 0 \quad \text{III-1}$$

En répétant cette procédure pour chacune des  $k$  chaînes du robot, nous obtenons un système de  $k$  équations reliant la position de la nacelle à la position des actionneurs. Chacune de ces équations est un polynôme du second degré en  $q_i$ . Ce polynôme possède soit :

- Deux solutions complexes conjuguées. Dans ce cas la position  $\mathbf{x}$  de la nacelle n'est pas accessible.
- Une solution réelle double. Dans ce cas, le mécanisme est dans une position singulière.
- Deux solutions réelles distinctes (position régulière).

Si la position  $\mathbf{x}$  est régulière, deux solutions réelles distinctes existent. Nous choisissons arbitrairement de ne conserver que la plus grande (pour obtenir la configuration correspondant à la plus petite, il suffit d'inverser le sens du vecteur directeur unitaire  $u_i$ )

Nous en déduisons le modèle géométrique inverse pour une position  $\mathbf{x}$  est:

$$q_i = (x - P_i B_i) u_i + \sqrt{[(x - P_i B_i) u_i]^2 - [(x - P_i B_i)^2 - l_i^2]} \quad \text{III-2}$$

### III.3.3.2. Modèle géométrique direct

Le modèle géométrique direct permet de calculer la position et l'orientation du point piloté de la nacelle, connaissant la position des actionneurs. Pour obtenir ce modèle, il faut résoudre le système (III-2) par rapport aux variables  $\mathbf{x}$  (aucune orientation dans l'architecture Delta). Dans le cas général, cette résolution ne peut se faire que de manière implicite car il n'existe pas de solution analytique à ce problème [25]. Nous remarquerons cependant que cette résolution est possible analytiquement, en particulier pour l'architecture Delta.

### III.3.4. Modélisation cinématique

Le modèle cinématique établit la relation entre la vitesse de la nacelle en translation et en rotation (que nous noterons  $\dot{\mathbf{x}}$ ) et  $\dot{\mathbf{q}}$  (vitesse linéaire des actionneurs) pour une position et une orientation données de la nacelle.

L'équiprojectivité des vitesses d'un solide est la propriété qui sera utilisée afin d'établir les modèles cinématiques. Cette propriété sera appliquée à chacune des  $k$  barres. Pour la barre numéro  $i$ , nous avons :

$$V_{A_i} \cdot A_i B_i = V_{B_i} \cdot A_i B_i \quad \text{III-3}$$

Avec :  $V_{A_i} = \dot{q}_i \cdot u_i$ ,  $V_{B_i} = V_c$  (car l'orientation est constante)

D'où  $\dot{q}_i \cdot u_i \cdot A_i B_i = \dot{\mathbf{x}} \cdot A_i B_i$

L'écriture pour l'ensemble des  $k$  barres nous donne l'écriture matricielle :

III-4

$$J_q \dot{q} = J_x \dot{x}$$

Avec :

$$J_q = \begin{bmatrix} A_1 B_1 \cdot u_1 & 0 & 0 \\ 0 & A_2 B_2 \cdot u_2 & 0 \\ 0 & 0 & A_3 B_3 \cdot u_3 \end{bmatrix}, \quad \text{III-5}$$

$$\text{et } J_x = \begin{bmatrix} (A_1 B_1)_x & (A_1 B_1)_y & (A_1 B_1)_z \\ (A_2 B_2)_x & (A_2 B_2)_y & (A_2 B_2)_z \\ (A_i B_i)_x & (A_i B_i)_y & (A_i B_i)_z \end{bmatrix} \quad \text{III-6}$$

Le modèle cinématique inverse est l'expression de  $\dot{q}$  en fonction de  $\dot{x}$ . L'écriture du modèle cinématique inverse à partir de l'équation (III-4) est alors :

$$\dot{q} = J_q^{-1} J_x \dot{x} \quad \text{III-7}$$

Soit en posant  $J = J_x^{-1} J_q$

$$\dot{q} = J^{-1} \dot{x} \quad \text{III-9}$$

Où  $J$  est appelée la matrice Jacobienne.

Le modèle cinématique direct est l'expression de  $\dot{x}$  en fonction de  $\dot{q}$ . Compte tenu des notations précédentes et de l'équation (III-8), le modèle cinématique direct s'écrit :

$$\dot{x} = J \dot{q} \quad \text{III-8}$$

### III.3.5. Choix d'un agencement

Maintenant que nous avons choisi et modélisé des points de vue géométrique et cinématique une architecture, il nous reste à répondre à deux questions : Comment allons nous disposer les glissières (topologie  $T$ ) et quelles dimensions (D) allons nous retenir? Souvent les outils utilisés en synthèse de mécanismes résolvent ces deux problèmes simultanément. Nous avons choisi de répondre à ces deux questions séparément l'une après l'autre, car, dans notre cas, le choix de la topologie relève de critères dont certains sont subjectifs et l'optimisation des dimensions de critères qualitatifs. Dans un premier temps nous allons déterminer quel est le meilleur type d'agencement et après nous rechercherons quelles sont les dimensions optimales pour cet agencement.

L'illustration de différents agencements est donnée sur la Figure III-3

Nous choisissons la solution « glissières parallèles à l'axe z » car l'encombrement de cette solution est faible et sa réalisation est plus simple que les autres.

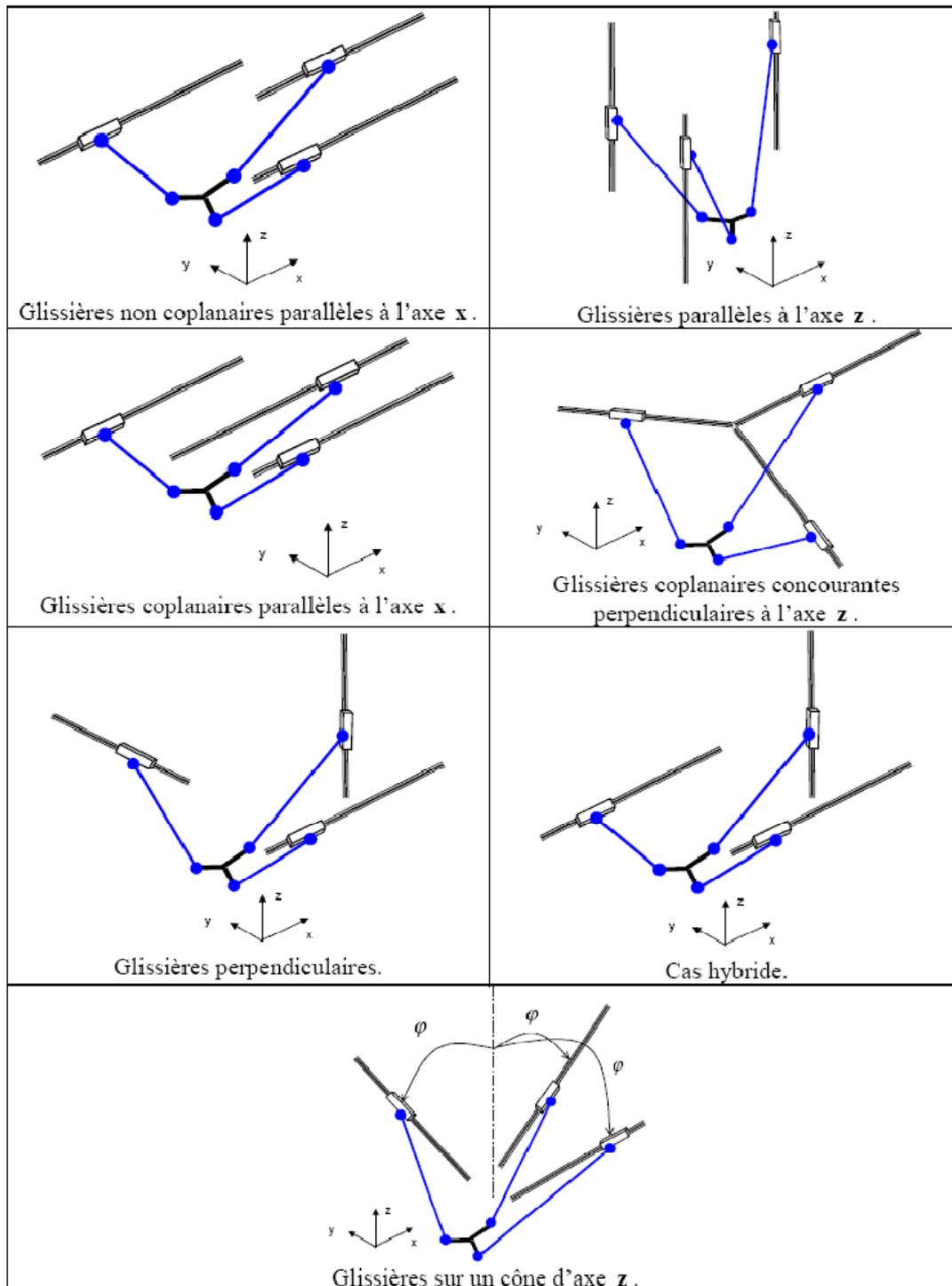


Figure III- 3 : Illustration des agencements étudiés.

### III.3.6. Réduction du nombre de paramètres.

La réduction du nombre de paramètres c.-à-d. une simplicité de réalisation de la machine, Par exemple, si les points  $P_i$  sont sur un cercle et espacés angulairement de manière régulière, nous n'utiliserons que deux paramètres pour décrire leur position : le rayon du cercle et la valeur angulaire de leur espacement. De même nous considérerons

que toutes les barres du mécanisme ont la même longueur  $L$ . Nous obtenons ainsi un nombre réduit de paramètres.

Nous réalisons une « contraction » de la nacelle. En effet, nous considérerons que la nacelle comme un disque de centre  $C$  et tous les points  $B_i$  sont placés sur la circonférence du disque. Et on considérera que chaque paire des barres comme une barre équivalente.

### III.4. Modélisation de l'architecture choisie.

#### III.4.1. Paramétrage.

L'illustration de l'architecture que nous avons choisie est représentée Figure III-4.

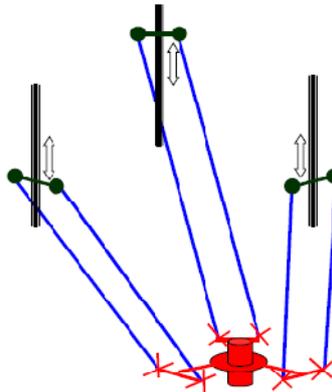


Figure III- 4 : Schéma de principe

Le paramétrage de cette agencement est présenté en Figure III-5. Les glissières sont disposées sur un cylindre de rayon  $R$ . La nacelle a pour rayon  $r$ . L'espacement angulaire des glissières est de 120 degrés.

#### III.4.2. Modèles géométriques

En utilisant les paramètres définis Figure III-5, les grandeurs intervenant dans les modèles génériques s'écrivent :

Coordonnées des points  $P_i$  dans le repère fixe :

$$[P_1 \quad P_2 \quad P_3] = \begin{bmatrix} R & -\frac{R}{2} & -\frac{R}{2} \\ 0 & R\frac{\sqrt{3}}{2} & -R\frac{\sqrt{3}}{2} \\ 0 & 0 & 0 \end{bmatrix} \quad \text{III- 10}$$

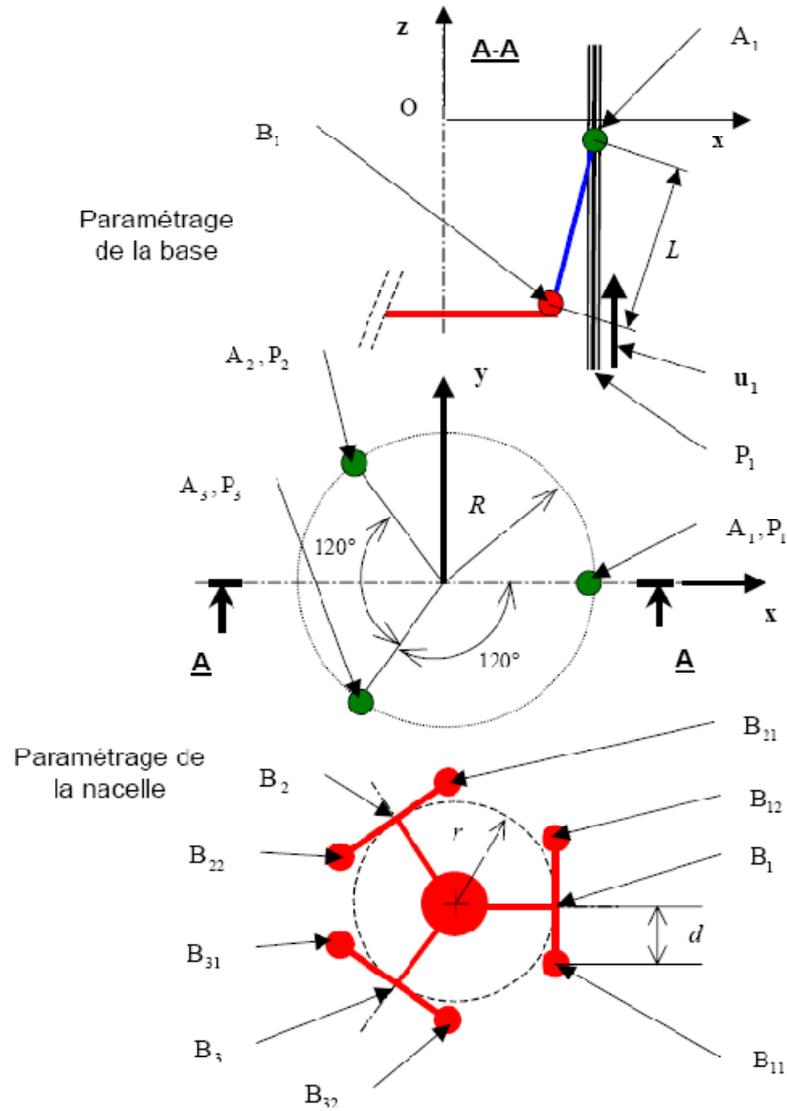


Figure III- 5 : Paramètres géométriques.

Coordonnées des points  $B_i$  dans le repère mobile  $R_n$  :

$$[B_1 \ B_2 \ B_3] = \begin{bmatrix} r & -\frac{r}{2} & -\frac{r}{2} \\ 0 & r\frac{\sqrt{3}}{2} & -r\frac{\sqrt{3}}{2} \\ 0 & 0 & 0 \end{bmatrix} \quad \text{III- 11}$$

Composantes des vecteurs  $u_i$  dans le repère fixe :

$$[u_1 \ u_2 \ u_3] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad \text{III- 12}$$

Nous obtenons l'expression analytique du modèle géométrique inverse :

$$\begin{cases} q_1 = z + \sqrt{L^2 - (r - R + x)^2 - y^2} \\ q_2 = z + \sqrt{L^2 - \left((R - r)\frac{1}{2} + x\right)^2 - \left((r - R)\frac{\sqrt{3}}{2} + y\right)^2} \\ q_3 = z + \sqrt{L^2 - \left((R - r)\frac{1}{2} + x\right)^2 - \left((R - r)\frac{\sqrt{3}}{2} + y\right)^2} \end{cases} \quad \text{III-13}$$

Pour obtenir l'expression analytique du modèle géométrique direct, nous devons résoudre le système (III-14) par rapport aux variables  $x$ ,  $y$  et  $z$ .

$$\begin{cases} (r - R + x)^2 + y^2 + (z - q_1)^2 = L^2 \\ \left((R - r)\frac{1}{2} + x\right)^2 + \left((r - R)\frac{\sqrt{3}}{2} + y\right)^2 + (z - q_2)^2 = L^2 \\ \left((R - r)\frac{1}{2} + x\right)^2 + \left((R - r)\frac{\sqrt{3}}{2} + y\right)^2 + (z - q_3)^2 = L^2 \end{cases} \quad \text{III-14}$$

Le système (III-14) peut se réécrire :

$$\begin{cases} (A^2 + C^2 + 1)z^2 + 2(AE + CD - q_1)z + E^2 + D^2 + q_1^2 - L^2 = 0 \\ y = Az + B \\ x = Cz + D \end{cases} \quad \text{III-15}$$

$$\text{Avec : } A = \frac{(q_2 - q_3)}{\sqrt{3}(r - R)} \quad B = \frac{q_3^2 - q_2^2}{\sqrt{3}(r - R)} \quad C = \frac{2(q_2 - q_1) - A\sqrt{3}(r - R)}{3(R - r)}$$

$$D = \frac{q_1^2 - q_2^2 - B\sqrt{3}(r - R)}{3(R - r)} \quad \text{et} \quad E = r - R + B$$

La première équation du système (III-15) admet deux solutions qui correspondent à deux positions de la nacelle. La solution correspondant à la machine étudiée est la position basse. Il faut donc conserver la plus petite des deux solutions de l'équation polynomiale. Connaissant  $z$ , nous pouvons ensuite calculer  $x$  et  $y$  de manière unique.

### III.4.3. Modèles cinématiques

Dans le cas de l'architecture choisie et avec le paramétrage de la Figure III-5, les matrices  $J_x$  et  $J_q$  de l'équation (III-7) s'écrivent :

$$J_x = \begin{bmatrix} r - R + x & y & z - q_1 \\ \frac{1}{2}(R - r) + x & \frac{\sqrt{3}}{2}(r - R) + y & z - q_2 \\ \frac{1}{2}(R - r) + x & \frac{\sqrt{3}}{2}(R - r) + y & z - q_3 \end{bmatrix} \quad \text{III-16}$$

$$\text{et } J_q = \begin{bmatrix} z - q_1 & 0 & 0 \\ 0 & z - q_2 & 0 \\ 0 & 0 & z - q_3 \end{bmatrix} \quad \text{III-17}$$

#### III.4.4. Modèles dynamiques.

Pour pouvoir écrire le modèle dynamique, nous établissons:

- La relation entre  $\ddot{\mathbf{q}}$  et  $\ddot{\mathbf{x}}$ .
- La relation entre un effort appliqué sur la nacelle et l'effort résultant sur les moteurs.
- Les masses en mouvement.

Nous recherchons l'expression de  $\ddot{\mathbf{q}}$  (accélération des moteurs) en fonction de  $\ddot{\mathbf{x}}$  (accélération désirée de la nacelle). En réalité est également fonction de  $\dot{\mathbf{x}}$ ,  $\dot{\mathbf{q}}$ ,  $\mathbf{x}$  et  $\mathbf{q}$ . La connaissance de cette relation nous permet de déterminer les caractéristiques requises concernant l'accélération des moteurs en fonction des performances en accélération demandées à la nacelle. L'expression recherchée s'obtient en dérivant le modèle cinématique par rapport au temps. En dérivant l'équation (III-4), nous obtenons:

$$\ddot{\mathbf{q}} = J^{-1}\ddot{\mathbf{x}} + J_q^{-1}(\dot{J}_x - \dot{J}_q J^{-1})\dot{\mathbf{x}}$$

Pour l'arrangement de l'architecture Delta que nous avons retenu, les matrices  $J_x$  et  $J_q$ , s'expriment analytiquement :

$$J_x = \begin{bmatrix} \dot{x} & \dot{y} & \dot{z} - \dot{q}_1 \\ \dot{x} & \dot{y} & \dot{z} - \dot{q}_2 \\ \dot{x} & \dot{y} & \dot{z} - \dot{q}_3 \end{bmatrix} \quad \text{et} \quad J_q = \begin{bmatrix} \dot{z} - \dot{q}_1 & 0 & 0 \\ 0 & \dot{z} - \dot{q}_2 & 0 \\ 0 & 0 & \dot{z} - \dot{q}_3 \end{bmatrix} \quad \text{III-19}$$

Nous constatons que l'accélération des moteurs est composée de la somme de deux termes :

- $J^{-1}\ddot{\mathbf{x}}$  L'accélération des actionneurs due à l'accélération de la nacelle.
- $(J_x - J_q J^{-1})\dot{\mathbf{x}}$  L'accélération des actionneurs due au déplacement de la nacelle à une vitesse constante. Ce terme traduit le non linéarité de la relation entre la vitesse des actionneurs et la vitesse de la nacelle.

Nous calculons dans un premier temps la relation qui relie l'effort de poussée des moteurs aux efforts appliqués sur la nacelle du point de vue statique. Les efforts appliqués sur la nacelle sont représentés par un torseur (Force et Moment) exprimé au point D. Comme le modèle cinématique est également exprimé au point D, nous pouvons en déduire (dualité entre les efforts et les vitesses due à la conservation de la puissance mécanique) la relation :

$$\mathbf{F}_{mot} = {}^t\mathbf{J} \mathbf{F}_{nac}$$

$$F_{nac} = \begin{bmatrix} F_x \\ F_x \\ F_z \end{bmatrix} \quad \text{et} \quad F_{mot} = \begin{bmatrix} F_{mot1} \\ F_{mot2} \\ F_{mot3} \end{bmatrix} \quad \text{III-21}$$

- $F_{mot}$  Effort de poussée des moteurs
- ${}^tJ$  Transposée de la matrice jacobienne définie dans modèle cinématique
- $F_{nac}$  Torseur des efforts appliqués sur la nacelle exprimé au point D.

➤ **Masses en mouvement.**

Pour établir le modèle dynamique inverse, nous devons prendre en compte les masses des pièces en mouvement. Les pièces en mouvement considérées sont :

- Nacelle (masse  $M_1$ ).
- Secondaire des moteurs linéaires (masse  $M_2$ ).
- Barres de fixation de la nacelle (masse  $M_b$ ).

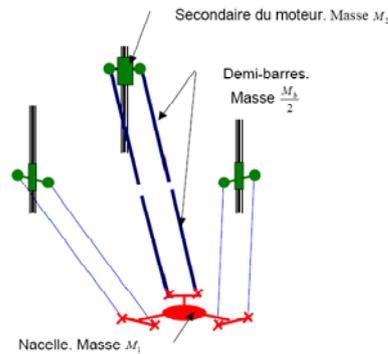
La pratique montre que, pour simplifier les calculs, la masse de chaque barre de fixation peut être répartie pour une moitié sur la nacelle et pour l'autre moitié sur le secondaire des moteurs, tandis que son inertie est négligée. Cette simplification est représentée sur la Figure III-6. La simplification proposée ci-dessus nous donne les masses corrigées (Annexe C) :

Masse corrigée de la nacelle :

$$M_{nac} = M_1 + 6 \frac{M_b}{2} = M_1 + 3M_b \quad \text{III-22}$$

Masse corrigée de chaque secondaire de moteur :

$$M_{mot} = M_2 + 6 \frac{M_b}{2} = M_2 + M_b \quad \text{III-23}$$



**Figure III- 6 : Répartition de la masse des barres.**

Le modèle dynamique inverse donne la poussée que doit fournir chacun des moteurs pour des vitesses et des accélérations données de la nacelle. Cette poussée des moteurs doit vaincre les forces de frottement (négligées dans cette étude), les efforts extérieurs

appliqués à la nacelle, l'inertie des moteurs en translation et l'inertie de la nacelle en translation.

- $\mathbf{M}_{mot}\ddot{\mathbf{q}}$  composante de l'effort moteur due à l'accélération de la masse ( $\mathbf{M}_{mot}$ ) de la partie mobile des moteurs (secondaire du moteur + liaison rotule). Tous les moteurs sont identiques (modularité). Qui peuvent s'écrire aussi :  $\mathbf{M}_{mot} \mathbf{J}^{-1}\ddot{\mathbf{x}} + \mathbf{M}_{mot} \mathbf{J}_q^{-1}(\dot{\mathbf{J}}_x - \dot{\mathbf{J}}_q \mathbf{J}^{-1})\dot{\mathbf{x}}$
- ${}^t\mathbf{J}\mathbf{F}_{ext}$  Composante de l'effort moteur due aux efforts extérieurs (principalement les efforts de coupe) appliqués à la nacelle.
- ${}^t\mathbf{J}\mathbf{F}_{nac,acc,D} = \mathbf{M}_{nac} {}^t\mathbf{J}\ddot{\mathbf{x}}$  : Composante de l'effort moteur due à l'accélération de la masse de la nacelle ( $\mathbf{M}_{nac}$ ) exprimé au point D.

Après regroupement, le modèle dynamique inverse s'écrit :

$$\mathbf{F}_{mot} = (\mathbf{M}_{nac} {}^t\mathbf{J} + \mathbf{M}_{mot} \mathbf{J}_q^{-1})\ddot{\mathbf{x}} + \mathbf{M}_{mot} \mathbf{J}_q^{-1}(\dot{\mathbf{J}}_x - \dot{\mathbf{J}}_q \mathbf{J}^{-1})\dot{\mathbf{x}} + {}^t\mathbf{J}\mathbf{F}_{ext} \quad III-24$$

➤ Efforts dans les barres.

Si nous isolons la nacelle, nous constatons qu'elle est soumise aux efforts suivants :

- Les efforts extérieurs (efforts de coupe)  $\mathbf{F}_{ext}, \mathbf{M}_{ext}$
- Les efforts dus à l'accélération de la masse de la nacelle  $\mathbf{F}_{acc}, \mathbf{M}_{acc}$
- Les efforts exercés par les 6 barres

Les efforts à l'intérieur d'une barre  $A_i B_i$  sont de type traction-compression (soit  $\mathbf{F}_{bi}$  le vecteur représentant cet effort) car les barres possèdent une liaison rotule à une extrémité et une liaison cardan à l'autre.

L'équilibre de la nacelle s'écrit :

$$\left\{ \begin{array}{l} \mathbf{F}_{ext} + \mathbf{F}_{acc} + \sum_{i=1}^3 \mathbf{F}_{bi} = \mathbf{0} \\ \mathbf{M}_{ext} + \mathbf{M}_{acc} + \sum_{i=3}^3 (\mathbf{F}_{bi} \times \mathbf{B}_i \mathbf{D}) = \mathbf{0} \end{array} \right. \quad III-25$$

Ce système d'équations s'écrit sous la forme matricielle :

$$\mathbf{J}_b \mathbf{F}_b = \begin{bmatrix} \mathbf{F}_{ext} \\ \mathbf{M}_{ext} \end{bmatrix} \quad III-26$$

Avec

$$J_b = \frac{1}{L} \begin{bmatrix} {}^t A_1 B_1 & {}^t (DB_{11} \times A_1 B_1) \\ {}^t A_1 B_1 & {}^t (DB_{12} \times A_1 B_1) \\ {}^t A_2 B_2 & {}^t (DB_{21} \times A_2 B_2) \\ {}^t A_2 B_2 & {}^t (DB_{22} \times A_2 B_2) \\ {}^t A_3 B_3 & {}^t (DB_{31} \times A_3 B_3) \\ {}^t A_3 B_3 & {}^t (DB_{32} \times A_3 B_3) \end{bmatrix} \quad \text{III-27}$$

Finalement, les efforts dans les barres sont donnés par :

$$F_b = J_b^{-1} \begin{bmatrix} F_{\text{ext}} \\ M_{\text{ext}} \end{bmatrix} \quad \text{III-28}$$

### III.5. Conclusion :

Dans ce chapitre, nous avons présenté la limite des machines outils d'architecture séries, et les avantages de l'architecture parallèle dans le domaine des machines outils.

Parmi les architectures parallèles connues aujourd'hui, nous avons exposé les architectures candidats comme une machine outil.

Nous avons choisi le robot delta et l'agencement correspondent. Et pour cet agencement nous avons fait une modélisation générique de la structure, et enfin la détermination de modèle géométrique, cinématique et dynamique correspondent. Il nous reste que l'optimisation de leurs dimensions qui se fait dans le chapitre suivant.

# Chapitre IV

## Optimisation multi-objectifs du robot Delta

### IV.1. Introduction :

Dans le chapitre précédent, nous avons sélectionné le type et la topologie de la machine. Il nous reste à effectuer une optimisation de quelque paramètre de système.

Dans ce chapitre nous proposons une approche basée sur les algorithmes génétiques (NSGA-II) pour la résolution du problème d'optimisation multi-objectif de notre machine.

Dans le cadre de notre travail, on utilise GEAToolbox qui possède l'algorithme génétique NSGA II pour une utilisation avec Matlab.

### IV.2. Description de la *Toolbox GEA* sous **MATLAB** :

The *Genetic and Evolutionary Algorithm Toolbox* pour une utilisation avec Matlab (**GEATbx**) mise en œuvre une grande gamme d'algorithmes génétiques et évolutionnaires pour résoudre les grands et complexes réels problèmes de monde. Par suite, le GEA Toolbox est un outil d'optimisation très puissant applicable à une large gamme de problèmes.

#### IV.2.1. Présentation de la structure de **GEATbx** :

Cette toolbox est formé d'un modèle de couches comme la figure

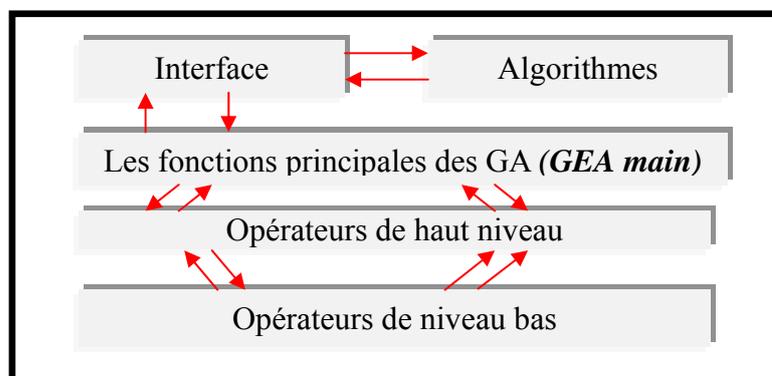


Figure IV- 1 : Architecture en couches du *GEATbx*

Le point central est la fonction principale « *geain2* », Cette fonction est appelée à partir de l'interface utilisateur (Demo fonctions, par exemple, *demofun1*, *demogeatbx*).

Les fonctions Toolbox définir des paramètres par défaut pour un certain nombre d'algorithmes évolutifs différents.

La fonction principale appelle tous les opérateurs évolutionnaires nécessaires. L'appelle des opérateurs évolutionnaires est faites par les opérateurs haut niveau, en appelant aussi les opérateurs de bas niveau. La fonction principale appelle aussi la fonction objective. En outre, la fonction Principale donne presque une représentation de toute la gestion de données et collection du résultat.

### IV.2.2. notations conventionnelles:

Dans la GEA toolbox, les fonctions sont nommées sont les conventions suivantes :

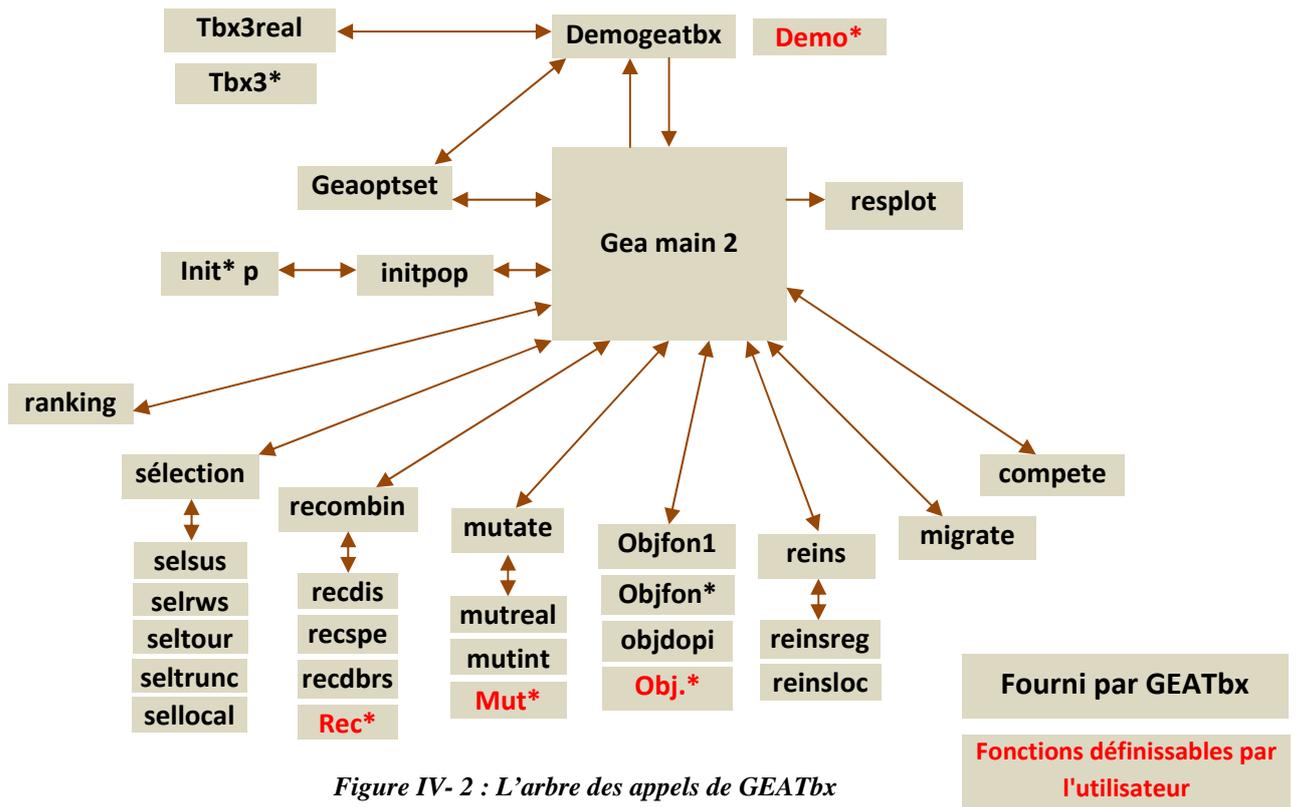
préfixe	Groupe de fonction et opérateur
<i>Démo</i>	Opérateur de haut niveau ( <i>demo</i> ) ensemble d'options pour un problème spécial, souvent utilisée comme démonstration
<i>Tbx3</i>	Opérateur de haut niveau <i>toolbox</i> algorithme Définir l'algorithme évolutionnaire a utilisé
<i>init</i>	<b>Initialisation</b> fonctions
<i>Sel</i>	Opérateur de <b>sélection</b> (de niveau bas)
<i>Rec.</i>	Opérateur de <b>recombinaison</b> (de niveau bas)
<i>Mut.</i>	Opérateur de <b>mutation</b> (de niveau bas)
<i>reins</i>	Opérateur de <b>réinsertion</b> (de niveau bas)
<i>Obj</i>	Fonctions <b>Objectives</b> (mettez en application un problème spécial)
<i>Mobj</i>	Fonctions <b>multi objective</b>
<i>sim</i>	Fonction de <b>simulation</b>
<i>Plot</i>	Fonction plot application spécifique

Table IV- 1 : Notation des fonctions dans GEATbx

### IV.2.3. L'arbre d'appel de GEATbx :

Le centre de la toolbox est la fonction principale « geamain2 », cette dernière est appelée par la fonction démo, cette fonction démo règle les paramètres de l'algorithme prédéfinie via toolbox fonction (*tbx3\**, *tbx3real*, *tbx3bin*, *tbx3es1*). La fonction principale appelle l'opérateur haut niveau (selection, recombinaison, mutata). L'opérateur haut niveau appelle l'opérateur bas niveau (selsus, recdis, mutreal).

Dans cette structure il y a deux points fondamentaux dans l'utilisation « input », la fonction objective et la fonction démo. Dans une partie l'utilisateur doit offrir une fonction objective pour implémenter son problème à résoudre, dans l'autre partie l'utilisateur doit définir un algorithme évolutionnaire pour employer un ou plusieurs options offertes par la toolbox.



#### IV.2.4. Démo/ fonction de démarrage :

Au départ il y a des fonctions hautes niveau pour définir quelques paramètres et leurs applications spécifiques. Ces fonctions sont appelées fonctions démo (*demo\*.m*).

**demofun1** : démo solution de *objfun1*, emploie des algorithmes évolutionnaires pour des paramètres de valeurs réels.

**demofun2** : solution démo, *objfun2*, emploie des algorithmes évolutionnaires avec ES comme étant un opérateur de mutation.

**demogeatbx** : menu simple permet d'utiliser 15 fonctions objectives et 3 différents algorithmes évolutionnaires.

Dans la fonction démo tous les paramètres de l'algorithme évolutionnaire et le processus d'optimisation peuvent être définies. Chaque aspect de comportement de GEA toolbox peut être contrôlé de l'extérieur via les ensembles des paramètres dans la fonction démo.

#### IV.2.5. Les fonctions de la toolbox (les algorithmes prédéfinis)

La GEA toolbox contient un certain nombre d'algorithmes évolutionnaires. Le processus de l'arrangement et les significations des fonctions, de ces algorithmes, de la toolbox sont donnés comme suit :

**Tbx3real** : pour paramètre réel

**Tbx3int** : pour les nombres entiers

**Tbx3bin** : variables évaluées en binaires

**Tbx3es1** : implémentation d'une stratégie évolutionnaire

**Tbx3comp** : l'émigration et compétition

**Tbx3perm** : implémentation de simple permutation

**Tbx3perm** : ordonnancement des algorithmes évolutionnaires.

**Tbx3sga** : implémente de simple algorithme génétique.

**Tbx3steadyga** : implémente d'état stable d'un algorithme génétique.

**Tbx3output** : combine un nombre d'option relevant des outputs textuels et graphiques.

L'utilisation de l'une de ces fonctions, les options respectives sont réalisées. Ceci facilite la lecture des fonctions de démo. Les options qu'ont été déjà définis dans les fonctions de toolbox peuvent être réutilisées par d'autres optimisations. D'autres paramètres peuvent être placés dans les fonctions de démo, même celles fournies par la fonction de la toolbox. Toutes les options non définies sont placées automatiquement à l'intérieur de la fonction principale `geamain2` en appelant le `geaoptset` pour vérifier la validité des options en même temps.

#### IV.2.6. Algorithmes évolutionniste – fonctions principales :

**geamain2** : (GEA toolbox fonction principale).

Cette fonction est la pilote principale pour GEATbx.

Ici tous les paramètres sont résolus et les fonctions appropriées sont appelées. En outre, les résultats sont rassemblés et les fonctions de plot sont appelées.

##### IV.2.6.1. Initialisation :

L'initialisation est donnée par :

**Initpop** : c'est une fonction d'initialisation de haut niveau. Il sert à initialiser la population entière, réorganiser les individus existant et manipuler certains d'autres aspects d'initialisation. Cette fonction appelle d'autres fonctions de niveau bas pour avoir des représentations spécifiques, ces fonctions sont :

- **Initrp** : initialisation de la population en utilisant des valeurs réelles.
- **Initip** : initialisation d'une population en utilisant des valeurs entières.
- **Initbp** : initialisation de la population en utilisant les valeurs binaires.

##### IV.2.6.2. La boucle générationnelle des EA :

Dès que la population des individus soit initialisée et évaluée, l'algorithme évolutionnaire sera exécuté selon le nombre de générations déjà définies, de nouvelles

populations sont produites et évaluées. Toutes les fonctions de l'algorithme évolutionnaire sont appelées par l'intermédiaire des fonctions de haut niveau. Parmi ces fonctions de haut niveau en trouvent :

- le rang de la fonction d'adaptation (ranking)
- la selection (selection )
- recombinaison et croisement (recombine)
- mutation (mutate)
- evaluation (obj\*.m)
- Reinsertion (reins)
- Migration (migrate)
- Competition (compete)
- Visualization (resplot)

#### IV.2.6.3. La sélection :

La fonction de haut niveau de la sélection est donnée comme suit :

**Sélection** : cette fonction appelle les fonctions de niveau bas pour utiliser le type de la sélection souhaité, ces fonctions de niveau bas de la sélection sont comme suit :

- **selsus** : la sélection par prélèvement stochastique universel
- **selrws** : sélection par la roue de la fortune
- **seltour** : sélection par tournoi

Il existe d'autres fonctions de niveau bas de la sélection, en peut citer par exemple sélection local mais les plus utilisées sont celles déjà cité en premier.

#### IV.2.6.4. Croisement

Pour le croisement, la fonction de haut niveau est donnée comme suit :

**Recombine** : Par cette fonction de haut niveau en fait appel à d'autres sous fonctions de croisement qu'en appellent les fonctions de niveau bas. Parmi ces fonctions de niveau bas les plus utilisées, on trouve :

1. utilisée pour tous les paramètres d'optimisation
  - **recdis** : croisement discret
2. Recombinaison pour les optimisations du paramètre de la valeur réelle
  - **Reclin** : croisement linéaire
  - **Recint** : croisement intermediaries
  - **Reclindex**: croisement linéaire étendue.

### 3. Recombinaison pour les optimisations du paramètre de la valeur binaire

- **Recdp** : croisement en deux points
- **Recsp** : croisement en un point.

Il existe d'autres fonctions de croisement, chacune a sa fonction et ces conditions d'utilisation. Le choix de la fonction de croisement nécessaire sera fait par la fonction de haut niveau.

#### IV.2.6.5. Mutation

Pour la mutation, la fonction de haut niveau est donnée par **mutate**.

Cette dernière appelle les fonctions de niveau bas pour la mutation et ça dépend de la nature des valeurs utilisées :

- valeurs réelles : **mutreal**
- les valeurs entières : **mutint**
- les valeurs binaires : **mutbin**

Pour l'évaluation stratégique de l'opération de mutation :

- **mutes1** : adaptation de la taille de l'étape de mutation
- **mutes2** : adaptation de la taille de l'étape de mutation des paramètres de stratégies

La mutation pour une optimisation de (permutation, ordonnancement, combinaison) tous sont implémentées dans « mutcomb » :

- **mutswap** : échange des variables
- **mutinvert** : partie inverse d'un individu.
- **mutmove** : mouvée une variable dans un individu.

#### IV.2.6.6. Evaluation

Les fonctions objectives dans la toolbox sont nommées « obj.\*.m ».

- Les algorithmes évolutionnaires utilisent les fonctions de test avec des représentations réelles. Ces fonctions sont données comme suit : **objfun1** et **objfun12**
- Pour la représentation binaire en utilise : **objone1**
- Pour l'optimisation dynamique en utilise : **objdopi** et **objling2**
- Fonctions de teste pour l'optimisation standard avec deux variables indépendantes sont : **objbran**, **objeaso**, **objgold** et **Objsixh**
- Un nombre de fonctions multi-objectives sont aussi données : **mobifonseca1**, **mobifonseca2**, **mobbelegundu**, **mobjdtlz1**, **mobjdtlz2**, **mobjdtlz3**, **mobjkita**, **mobiquagliarella**, **mobcantilever**, **mobdebconstr** et **mobjsoland**.

#### IV.2.6.7. Réinsertion

Pour la réinsertion la toolbox offert des fonctions de haut niveau:

- **reins** : (réinsertion de saut externe de la population).
- **reinsreg** : (réinsertion global /régional de saut extérieur dans la population)
- **reinsloc** : (réinsertion local de saut extérieur dans la population)

#### IV.2.6.8. Migration

Pour l'émigration, la toolbox possède une seule fonction de haut niveau et sans fonctions de niveau bas. Cette fonction de haut niveau est **migrate**.

#### IV.2.6.9. Compétition

Pour la compétition, la toolbox contient aussi une seule fonction de haut niveau et sans fonction de niveau bas. Cette fonction de haut niveau est **compete**. Cette fonction est utilisée pour la compétition entre sous population.

#### IV.2.6.10. La visualisation

La visualisation des résultats se fait par 3 méthodes différentes :

- des résultats donnés sous forme graphique, donne par **resplot**
- des résultats donnés sous forme tableau, donne par **resplot**
- résultats spécifiés de problème sont donnés par **plotdopi**

#### IV.2.6.11. Les fonctions auxiliaires

Parmi les fonctions auxiliaires qu'on trouve dans la toolbox sont :

- **compdiv** : utiliser pour divers calcul.
- **chkbound** : contrôle des limites et remet des variables en dehors des limites
- **savebindata** : économiser les variables de matlab en utilisant un nom fourni à un dossier mat.
- **expandm** : utiliser pour augmenter une matrice.
- **straddname/straddtime** : ajouter un nom à une chaîne ou le temps courant à une chaîne.
- **deblankall** : débloquent les frontières d'une chaîne.
- **geaoptset** : placer une option de GEAOpt.

### IV.3. Formulation du problème multi-objectif :

#### IV.3.1. Les fonctions objectives :

Plusieurs critères ont été proposés afin de comparer différents mécanismes par rapport à leur géométrie, leur architecture ou leurs dimensions. Ces critères peuvent être de plusieurs natures :

- Géométrique (encombrement de la machine, course des actionneurs, ratio encombrement de la machine par rapport à son volume de travail, volume de travail)
- Cinématiques (transformation des vitesses, isotropie, absence de singularités)
- Dynamiques (poussée des moteurs, efforts dans la structure, capacité d'accélération)
- Autres (rigidité de la machine, précision, facilité d'étalonnage)

Nous allons choisir deux critères cinématiques qui sont respectivement le conditionnement de la matrice Jacobienne et la capacité de l'accélération de la nacelle.

#### IV.3.1.1. Accélération de la nacelle:

D'après l'équation (III-18) :

$$\ddot{\mathbf{q}} = \mathbf{J}^{-1}\ddot{\mathbf{x}} + \mathbf{J}_q^{-1}(\dot{\mathbf{J}}_x - \dot{\mathbf{J}}_q\mathbf{J}^{-1})\dot{\mathbf{x}}$$

D'où

$$\ddot{\mathbf{x}} = \mathbf{J}(\ddot{\mathbf{q}} - \mathbf{J}_q^{-1}(\dot{\mathbf{J}}_x - \dot{\mathbf{J}}_q\mathbf{J}^{-1})\dot{\mathbf{x}}) \quad \text{IV-1}$$

Dans notre programme nous minimisons  $1/\ddot{\mathbf{x}}$ , pour trouver le maximum de  $\ddot{\mathbf{x}}$

#### IV.3.1.2. Conditionnement de la matrice Jacobienne :

La définition de ce critère, également appelé indice d'isotropie, est le conditionnement de la matrice Jacobienne, soit :

$$c = \text{cond}(\mathbf{J}) = \frac{\sigma_{max}}{\sigma_{min}} \quad \text{IV-2}$$

La matrice  $\mathbf{J}$  transforme les vitesses de l'espace articulaire en vitesses de l'espace cartésien. Si le lieu des vitesses articulaires possibles est une hyper-sphère, le lieu des vitesses cartésiennes correspondantes est alors un hyper-ellipsoïde. Le critère  $c$  est le rapport entre le grand axe et le petit axe de cet hyper-ellipsoïde. Cette constatation permet de relier ce critère à l'isotropie cinématique de la machine.

Le conditionnement de la matrice Jacobienne est invariant pour des machines homothétiques. Il est toujours plus grand que 1. Plus le conditionnement de la matrice Jacobienne est proche de 1 implique que l'isotropie de la machine est meilleure.

### IV.3.2. Variables et contraintes :

#### IV.3.2.1. Limite de l'espace de travail et variables de conception

- Limite de l'espace de travail  $x$  et  $y$  (m)

$$X_{es} = \begin{cases} -0.25 \leq x \leq 0.25 \\ -0.25 \leq y \leq 0.25 \end{cases}$$

- Limites des variables de conception

$$(X_{c,i})_{min} \leq X_{c,i} \leq (X_{c,i})_{max}, \quad i = 1, \dots, n \quad \text{IV-4}$$

$$X_c = \{L, R, r, D\}$$

- ❖  $L$  : Longueur des barres
  - ❖  $R$  : Rayon de la base fixe
  - ❖  $r$  : Rayon de la nacelle
  - ❖  $D_b$  : Diamètre de section des barres
- Limites de forces extérieures :  
Les forces extérieures agissent sur la nacelle, qui sont les forces de coupes, ne dépassent pas une limite donnée.

$$|(F_{ext})_j| \leq (F_{ext})_{max}, \quad j = 1, 2, 3$$

- Limites cinématiques articulaires :

$$|\dot{q}_j| \leq (\dot{q}_j)_{max}, \quad j = 1, 2, 3$$

$$|\ddot{q}_j| \leq (\ddot{q}_j)_{max}, \quad j = 1, 2, 3$$

Nous pouvons définir le vecteur de variables  $X_T = \{X_c, F_{ext}, \dot{q}, \ddot{q}\}$

#### IV.3.2.2. La rigidité des barres :

Comme nous avons présenté précédemment que toutes les articulations sont considérées parfaitement rigides.

Par construction, les barres sont sollicitées uniquement en traction-compression. Ces sollicitations restent suffisamment petites pour que les problèmes liés au flambement ne soient pas présents. La déformation d'une de ces barres est :

$$Df_b = \frac{F_b L}{SE}$$

Avec :

- $F_b$  : Effort de traction-compression dans la barre (l'équation III-27)
- $L$  : Longueur initiale de la barre
- $S$  : Surface d'une section droite de la barre
- $E$  : Module d'Young du matériau dans lequel sont réalisées les barres

Cette déformation ne faut pas dépasser une valeur admissible trop petite  $Df_b \leq (Df_b)_{ad}$  pour éviter la grande erreur de déplacement de la nacelle, qui influé négativement sur la rigidité de la machine.

#### IV.3.2.3. L'absence de la configuration singulière :

Les configurations singulières sont des postures particulières de l'organe terminal où la rigidité naturelle des manipulateurs parallèles subit une grande détérioration.

On peut aussi introduire les singularités en abordant sommairement la notion d'équilibre mécanique d'un robot parallèle. Pour un manipulateur parallèle nous notons  $\boldsymbol{\tau}$  le vecteur des forces articulaires et  $\boldsymbol{\mathcal{F}}$  le torseur des efforts externes appliquées sur l'organe terminal. Pour un torseur  $\boldsymbol{\mathcal{F}}$  appliqué sur le plateau mobile, le système mécanique est en équilibre s'il existe des forces articulaires dont l'action sur la plateforme est l'opposée de  $\boldsymbol{\mathcal{F}}$ . Si ce n'est pas le cas, l'organe terminal du manipulateur va se déplacer jusqu'à ce qu'une nouvelle position d'équilibre soit atteinte. Or il existe une relation bien connue entre  $\boldsymbol{\tau}$  et  $\boldsymbol{\mathcal{F}}$  :

$$\boldsymbol{\mathcal{F}} = \boldsymbol{J}^{-t} \boldsymbol{\tau}$$

Où  $\boldsymbol{J}^{-t}$  est la transposée de la matrice Jacobienne cinématique inverse. L'équation précédente décrit un système linéaire en terme de composantes du vecteur  $\boldsymbol{\tau}$  qui admettra en général une solution en  $\boldsymbol{\tau}$  pour tout  $\boldsymbol{\mathcal{F}}$  (solution qui conduit donc à un équilibre mécanique du système) sauf dans le cas où la matrice  $\boldsymbol{J}^{-t}$  est dégénérée : dans ce cas le système linéaire n'admet pas de solution et le système mécanique n'est plus en équilibre.

Donc on prend comme contrainte :

$$|\boldsymbol{J}^{-t}| \neq 0$$

#### IV.3.2.4. Forces des Moteur :

Les forces fournies par les moteurs doivent être compatibles avec les limites de performances attendues (vitesses, accélérations et charge autorisées) en tout point de l'espace de travail.

L'expression des forces des moteurs est donnée par (III- 24) :

$$\boldsymbol{F}_{mot} = (\boldsymbol{M}_{nac} \quad {}^t\boldsymbol{J} + \boldsymbol{M}_{mot} \boldsymbol{J}_q^{-1}) \ddot{\boldsymbol{x}} + \boldsymbol{M}_{mot} \boldsymbol{J}_q^{-1} (\dot{\boldsymbol{J}}_x - \dot{\boldsymbol{J}}_q \boldsymbol{J}^{-1}) \dot{\boldsymbol{x}} + {}^t\boldsymbol{J} \boldsymbol{F}_{ext}$$

Donc :

$$\boldsymbol{F}_{mot} \leq (\boldsymbol{F}_{mot})_{ad}$$

### IV.4. Procédure algorithmique développées

La figure (IV-3) représente l'approche numérique que nous avons utilisée pour résoudre notre problème d'optimisation multi-objectif.

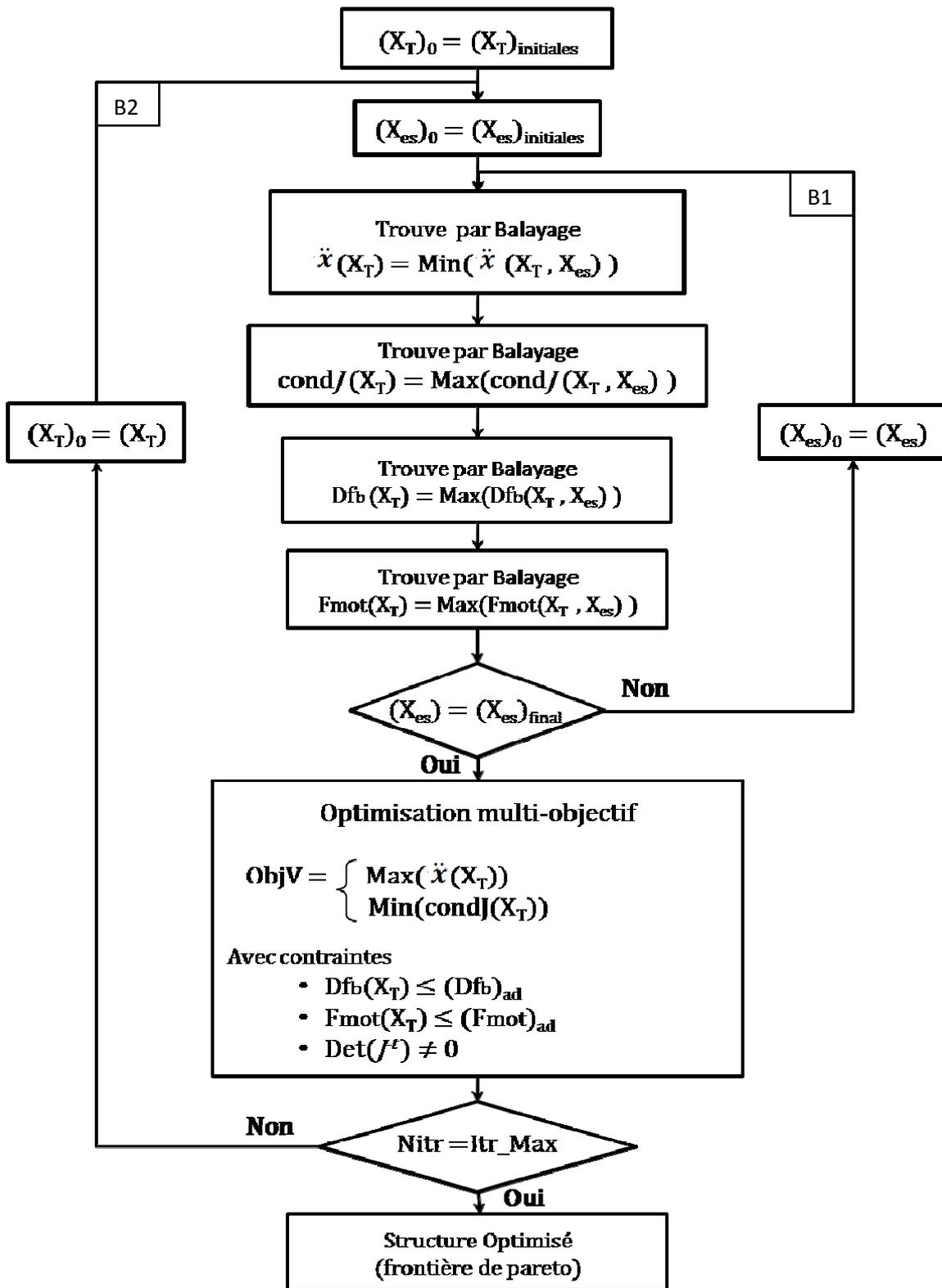


Figure IV- 3 : Schéma de résolution

Dans la boucle (B1), l'algorithme donne, pour une valeur de  $X_c$  (paramètres de conception), des valeurs optims de l'accélération de centre de la nacelle  $\ddot{x}$  et le conditionnement de la matrice Jacobienne  $cond(J)$ , le maximum de déformation des barres et la force maximale fournie par les moteurs dans toute l'espace de travail.

La boucle (B2) traite le problème d'optimisation multi-objectif. La résolution de ce problème permis de trouver des valeurs des paramètres de conception maximisant l'accélération de centre de la nacelle  $\ddot{x}$ , et minimisant le conditionnement de la matrice Jacobienne  $cond(J)$  simultanément, tant que les contraintes sont vérifiées. L'algorithme génétique pour l'optimisation multi-objectif donne un ensemble des solutions qui forment la frontière de Pareto.

#### IV.4.1. Les fonctions MATLAB :

La Première fonction assure l'exécution de la boucle (B1), et donne l'évolution des fonctions objectives.

La deuxième fonction contient :

- La fonction principale (geamain) ; Tout le travail administratif est effectué par cette fonction, parmi ces travaux administratifs, on trouve ; résoudre les valeurs des paramètres, l'initialisation de la population, l'exécution de l'algorithme évolutionnaire comprenant l'appel de la fonction objective et l'affichage et enregistrement des résultats.
- Présentation des paramètres de GEATbx.
- L'espace de recherche ; les bornes de limites des variables.

La boucle (B2) est traitée dans cette fonction.

#### IV.4.2. Paramétrage :

- Choix d'une population initiale (Nombre d'individus) ;
- Choix de sous population ;
- Définition des opérateurs génétiques ;
  - ✓ Sélection (type de sélection) ;
  - ✓ Croisement (type de croisement) ;
  - ✓ Mutation (type de mutation) ;

#### IV.4.3. Données de l'application :

##### IV.4.3.1. Données de la méthode de balayage :

Dans la boucle B1 (Figure IV-3) nous avons utilisé la méthode de balayage pour trouver l'optimum de chaque paramètre du problème défini. Nous discrétisons le volume de travail en 51 points sur x et y, c.-à-d. 2601 points dans tout le domaine.

## IV.4.3.2. Données de l'algorithme NSGA II :

Paramètres	Valeurs
Taille de la population	200 Individus
Sous population	8
Nombre de générations	700 générations
Nombre de fonctions objectives	02 fitness
Nombres de contraintes	03
Nombre de variable	13 variables
Pression de sélection	1.9
Type de mutation	mutation real
Probabilité de mutation	0.2
Type de codage	Codage real

Table IV- 2 : données de l'algorithme NSGA II

## IV.4.3.3. Caractéristique du robot :

	Paramètre	Valeur	
		Min	Max
Espace de recherche	$x$ (m)	-0.25	0.25
	$y$ (m)	-0.25	0.25
	$R$ (m)	0.5	0.7
	$r$ (m)	0.1	0.3
	$L$ (m)	0.8	1.7
	$D_b$ (m)	0.01	0.02
	$ \dot{q}_j _{max}$ (m/s)	3	
	$ \ddot{q}_j _{max}$ (m/s <sup>2</sup> )	35	
	$(F_{ext})_{max}$ (N)	<b>1000</b>	
Paramètres de robot	(M <sub>1</sub> ) Masse de Nacelle (kg)	20	
	(M <sub>2</sub> ) Masse de Moteur (kg)	10	
	Masse volumique des barres (kg/m <sup>3</sup> )	7849	
	Module d'élasticité des barres (N/mm <sup>2</sup> )	210000	
	$(Df_b)_{ad}$ (mm)	1	
	$(F_{mot})_{ad}$ (N)	9000	

Table IV- 3 : Caractéristique du robot

## IV.5. Résultats et discussions

D'après l'exécution du programme de l'algorithme génétique NSGA II sous Matlab, donne les résultats graphiques qui sont présentées dans les figures (IV-3), Ces résultats graphiques contiennent :

- La frontière de Pareto en 2D qui représente la solution de la meilleure itération.
- La représentation de l'espace de recherche en 3D de la meilleure itération.

- Graphe représente les valeurs des variables des meilleurs objectifs en fonction des générations.
- Graphe représente les valeurs des objectifs en fonction des générations.
- L'évolution des variables individuels à chaque itération.

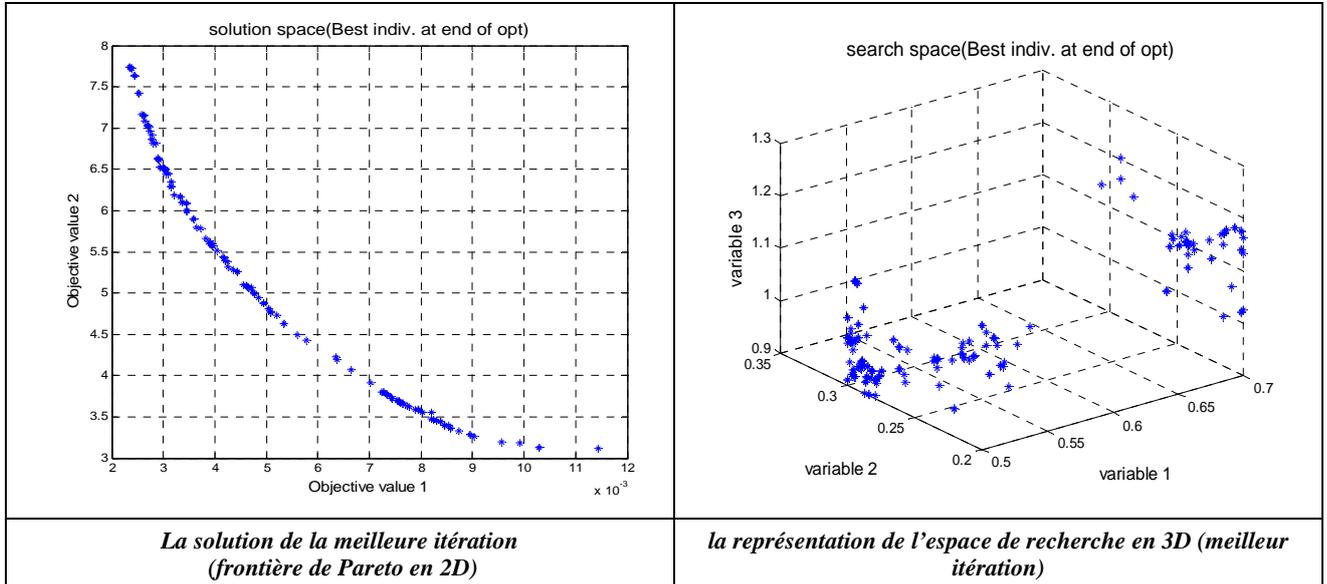


Figure IV- 4 : résultat graphique par GEATbx

Nous avons résumé les résultats obtenues dans (Table IV-4) qui représente les valeurs de meilleures solutions des fonctions objectifs ( $\ddot{x}$  et  $Cond(J)$ ) et les variables ( $R, r, L, D_b$ ). Ce résultat est un choix d'une solution parmi les résultats représentés dans la frontière de Pareto.

Fonctions Objectives	
$\ddot{x} \left(\frac{m}{s^2}\right)$	$Cond(J)$
225.337	5.259

Paramètres de conception			
$R$ (m)	$r$ (m)	$L$ (m)	$D_b$ (m)
0.519	0.218	1.100	0.011

Paramètres cinématiques						Forces extérieures		
$\dot{q}_1 \left(\frac{m}{s}\right)$	$\dot{q}_2 \left(\frac{m}{s}\right)$	$\dot{q}_3 \left(\frac{m}{s}\right)$	$\ddot{q}_1 \left(\frac{m}{s^2}\right)$	$\ddot{q}_2 \left(\frac{m}{s^2}\right)$	$\ddot{q}_3 \left(\frac{m}{s^2}\right)$	$(F_{ext})_x$ (N)	$(F_{ext})_y$ (N)	$(F_{ext})_z$ (N)
3	-3	-2.773	35	35	-35	-782.24	-606.72	-553.22

Table IV- 4 : Solution choisie

D'après de trouver les valeurs des paramètres géométriques de robot, nous avons représenté l'évolution des certaines grandeurs caractéristiques (Accélération de centre de nacelle, Conditionnement de la matrice Jacobienne, Déformation des barres et poussé des moteurs), Comme il n'existe pas de solution analytique pour l'évaluation de ces grandeurs, nous effectuerons un balayage de l'espace de travail (intérieur d'un cube de 0,5m de côté) selon  $x$  et  $y$  (nous n'effectuons pas de balayage suivant  $z$  car la géométrie

et donc les caractéristiques de la machine sont invariantes quand  $z$  change). Les résultats sont présentés dans les figures suivantes.

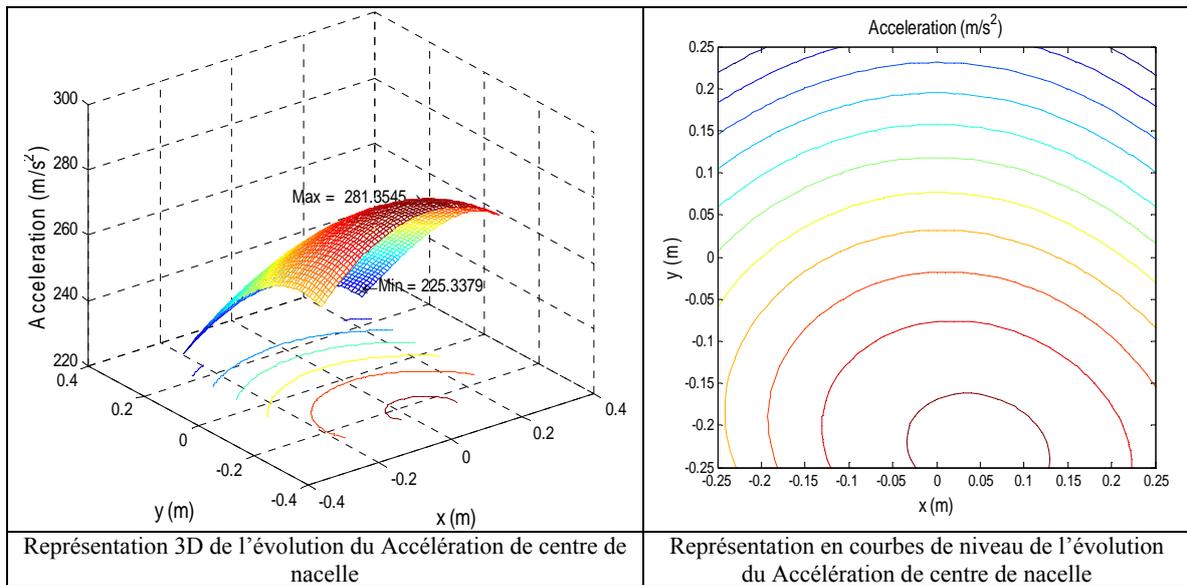


Figure IV- 5 : Evolution du Accélération de centre de nacelle

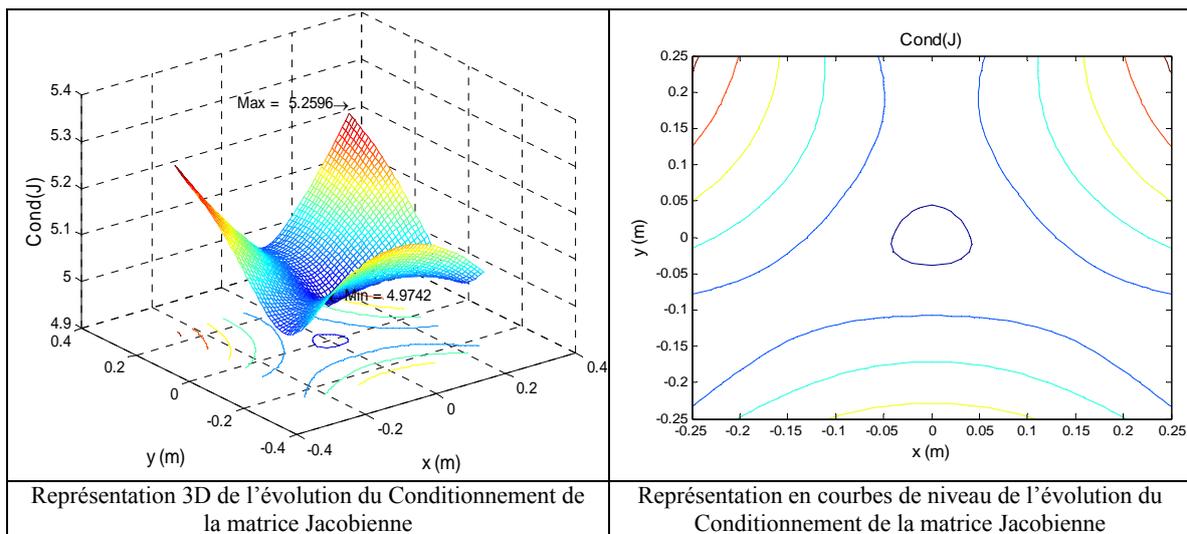


Figure IV- 6 : Evolution du Conditionnement de la matrice Jacobienne

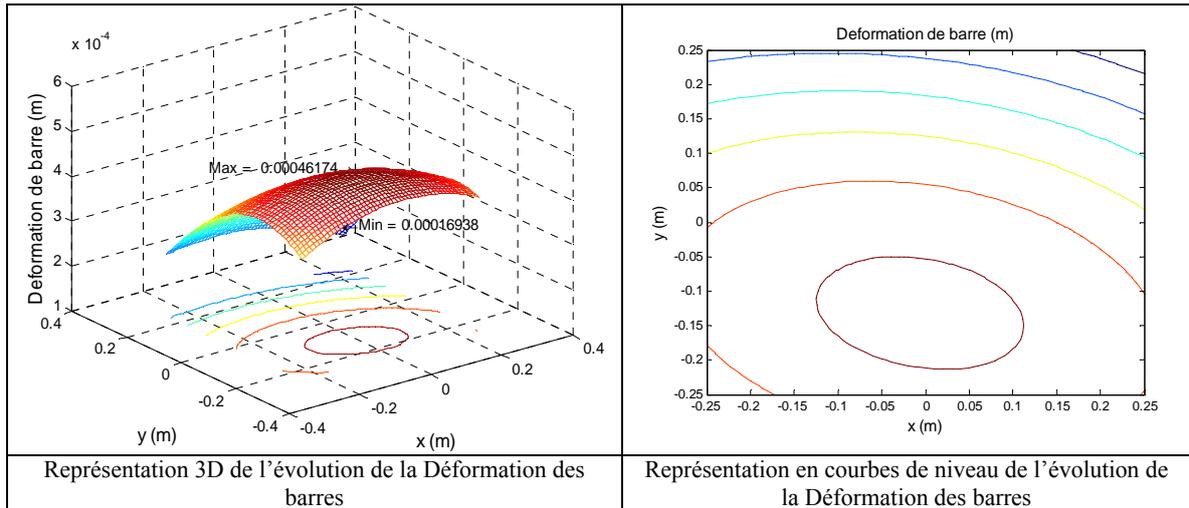


Figure IV- 7 : Evolution de la Déformation des barres

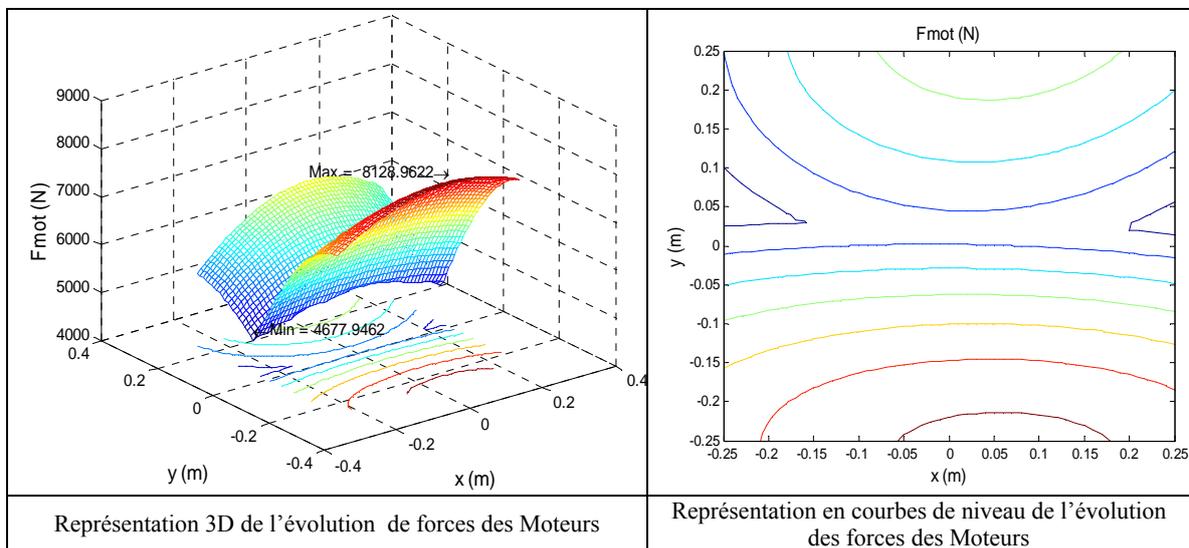


Figure IV- 8 : Evolution des forces des Moteurs

Le tableau (IV-5) représente la valeur maximale de la force appliquée par les moteurs, et la plus grande déformation de barres dans une tranche de volume de travail.

	$(F_{mot})_{max} = 8128,962 \text{ N}$	$(Df_b)_{max} = 0,0004617 \text{ m}$
$x \text{ (m)}$	0.06	-0.25
$y \text{ (m)}$	-0.01	-0.13

Table IV- 5 : Valeurs maximales de Fmot et Df<sub>b</sub>

## IV.6. Conclusion

Dans cette partie, nous avons présenté la « GEATbx » (*Genetic and Evolutionary Algorithm Toolbox*) que nous avons utilisé pour optimiser notre problème d'optimisation multi-objectif. Nous avons formulé le problème d'optimisation multi-objectifs, d'après définir tout les paramètres de problème, et les donnés par rapport le robot Delta. L'utilisation de l'algorithme génétique NSGA II nous donne les résultats présentés d'avant. L'optimisation de notre machine nous donne les dimensions optimales que nous allons utiliser dans le chapitre suivant pour simuler numériquement notre robot Delta.

# Chapitre V

## Simulation numérique

### V.1. Introduction

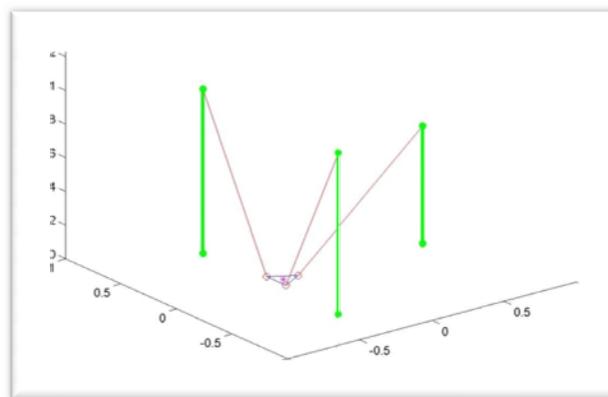
L'optimisation qui se fait dans le chapitre précédent, nous permet de choisir la dimension de notre robot d'après les valeurs des paramètres géométriques qui sont exportées du programme de GEATbox.

Dans cette partie, nous allons faire une évolution de quelques paramètres de robot, avec MATLAB et avec logiciel de CAO qui est « SolidWorks ».

### V.2. Structure optimal retenue

Le dimensionnement géométrique de robot est choisi d'après les résultats obtenus dans le chapitre précédent.

Nous avons présenté le robot Delta avec MATLAB (Figure V-1), et avec SolidWork (Figure V-2).



*Figure V-1 : Représentation de robot Delta sous MATLAB*

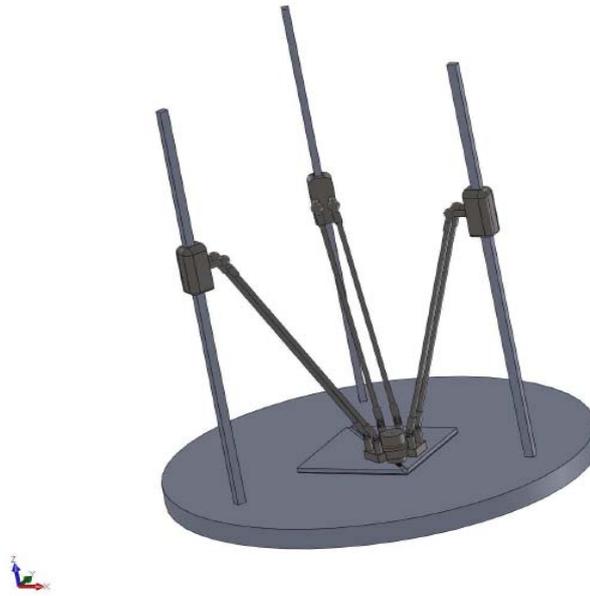


Figure V- 2 : Représentation de robot Delta sous SolidWork

### V.3. Simulation du Robot Delta

#### V.3.1. Condition de la simulation

Nous avons choisi les dimensions de notre robot et représenté leur structure, dans cette partie nous allons voir l'évolution de quelque paramètre de robot tel que les déplacements des moteurs, leurs vitesses et leurs accélérations, avec une trajectoire de centre de la nacelle, leur vitesse et leur accélération sont donnés.

La figure V-3 représente les différentes étapes de la démarche de notre travail, de modélisation de notre système (Robot Delta) jusqu'à l'évolution des paramètres sorties.

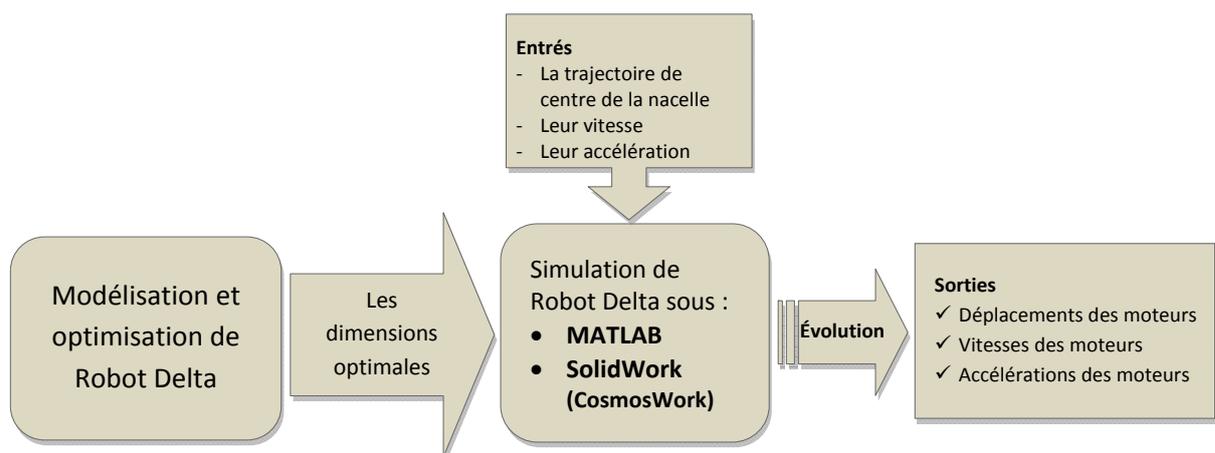


Figure V- 3 : Les différentes étapes de notre travail

#### V.3.2. Simulation du Robot Delta sous MATLAB

Nous avons choisi une trajectoire du centre de la nacelle avec une fonction  $y = f(x)$  avec  $x$  une fonction de temps, et nous voyons l'évolution des déplacements des trois moteurs, ses vitesses, et ses accélérations avec un programme implanté sous MATLAB.

c) Cas  $n^{\circ}=1$

$$x = 0.1 * t - 0.25$$

$$y = 0.25 * \sin(50 * x);$$

Quelques instants de la simulation représentent dans les figures suivantes.

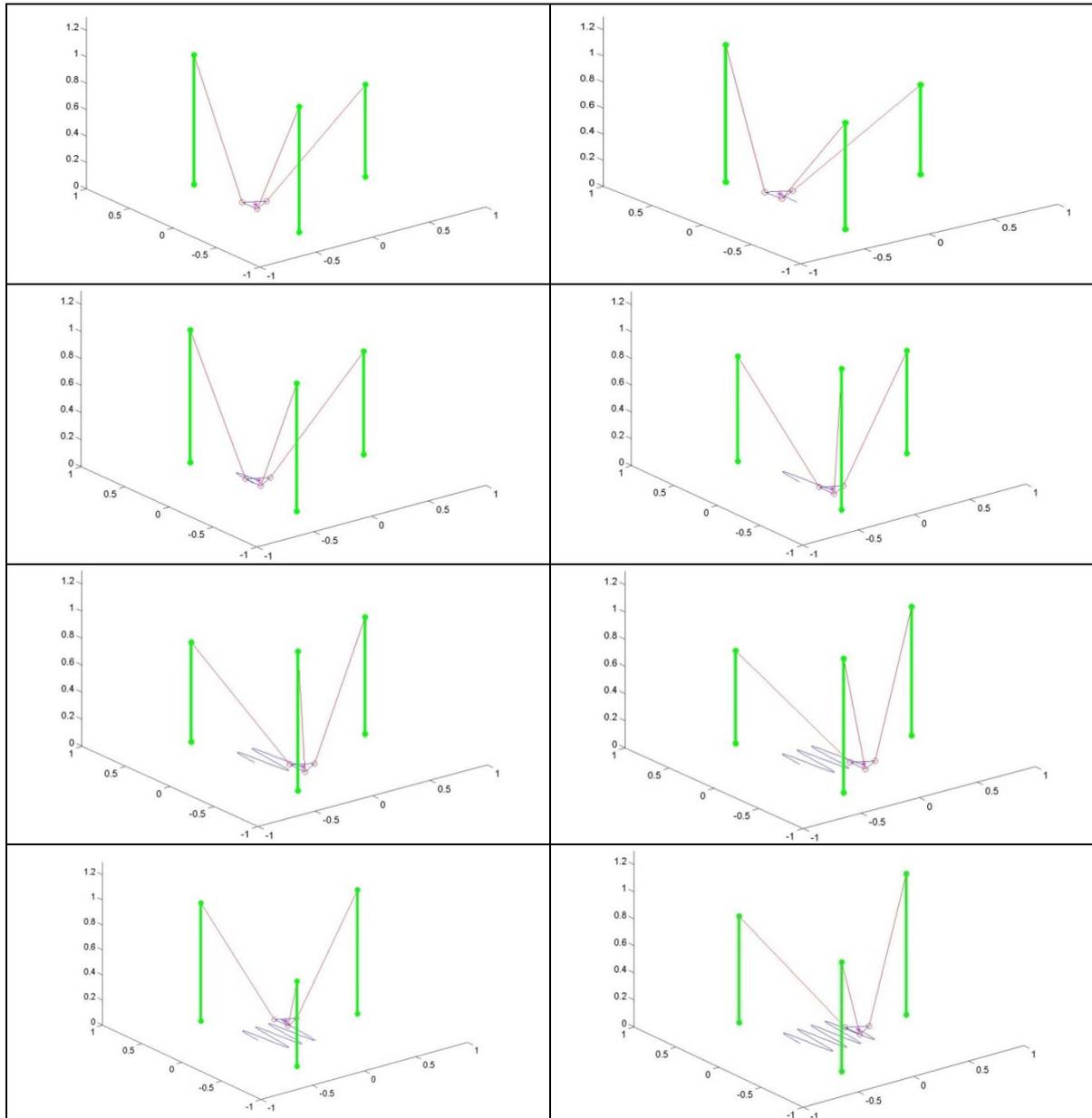


Figure V- 4 : Des instants de 1<sup>ère</sup> cas

➤ Les Entrés :

## 1. Trajectoire de centre de la nacelle

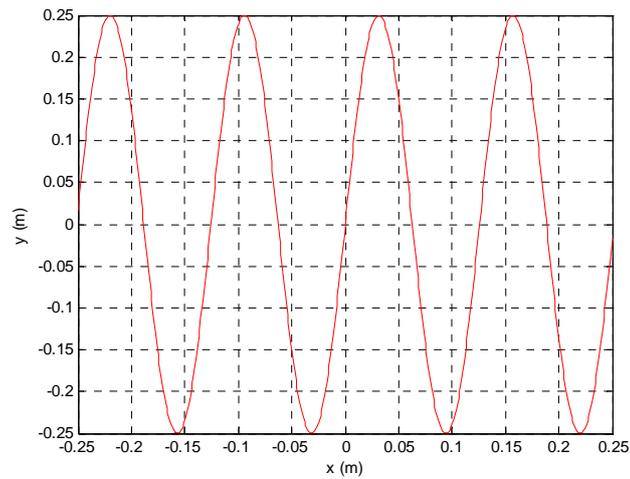


Figure V- 5 : Trajectoire de centre de la nacelle (1<sup>ère</sup> cas)

## 2. Vitesse de centre de la nacelle

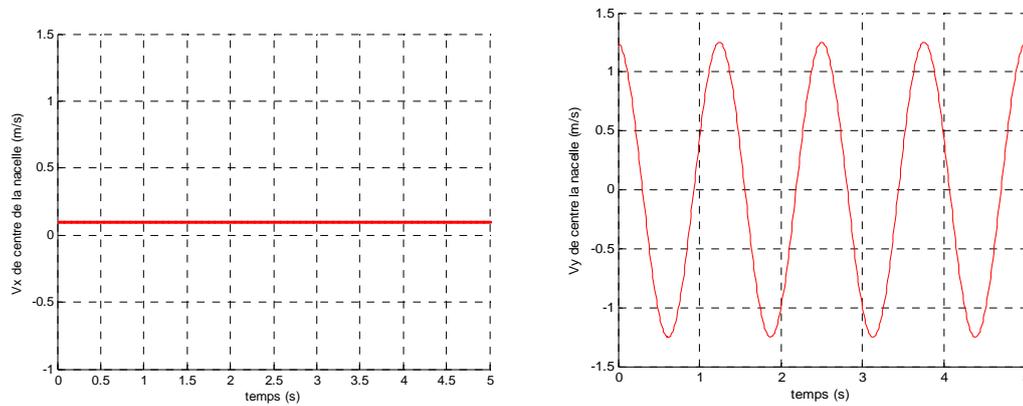


Figure V- 6 : Vitesse de centre de la nacelle (1<sup>ère</sup> cas)

## 3. Accélération de centre de la nacelle

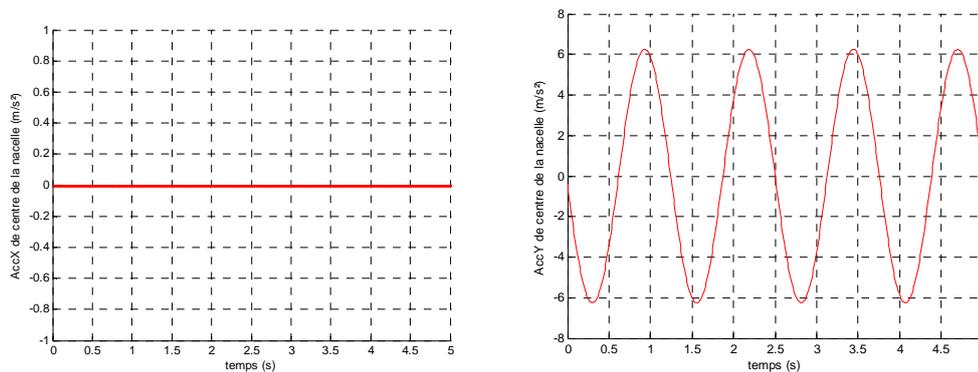


Figure V- 7 : Accélération de centre de la nacelle (1<sup>ère</sup> cas)

## ➤ Les Sortie

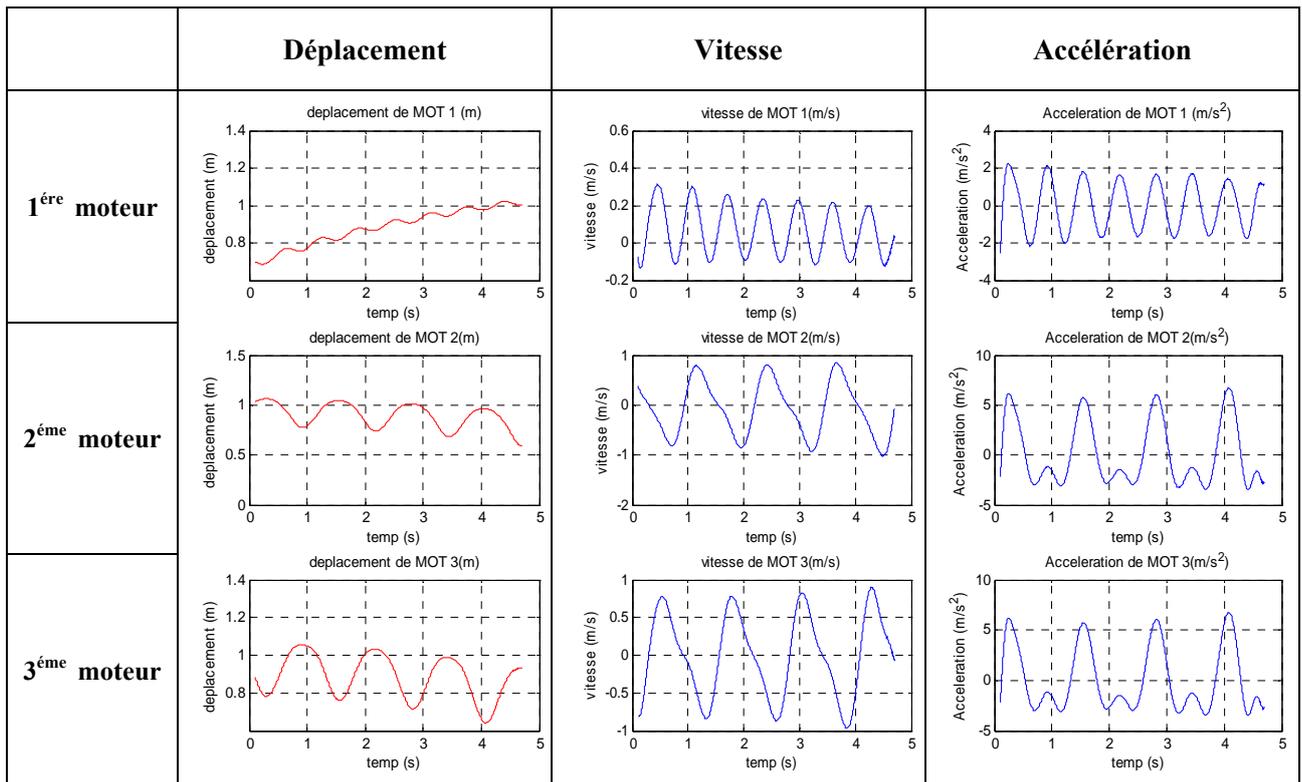
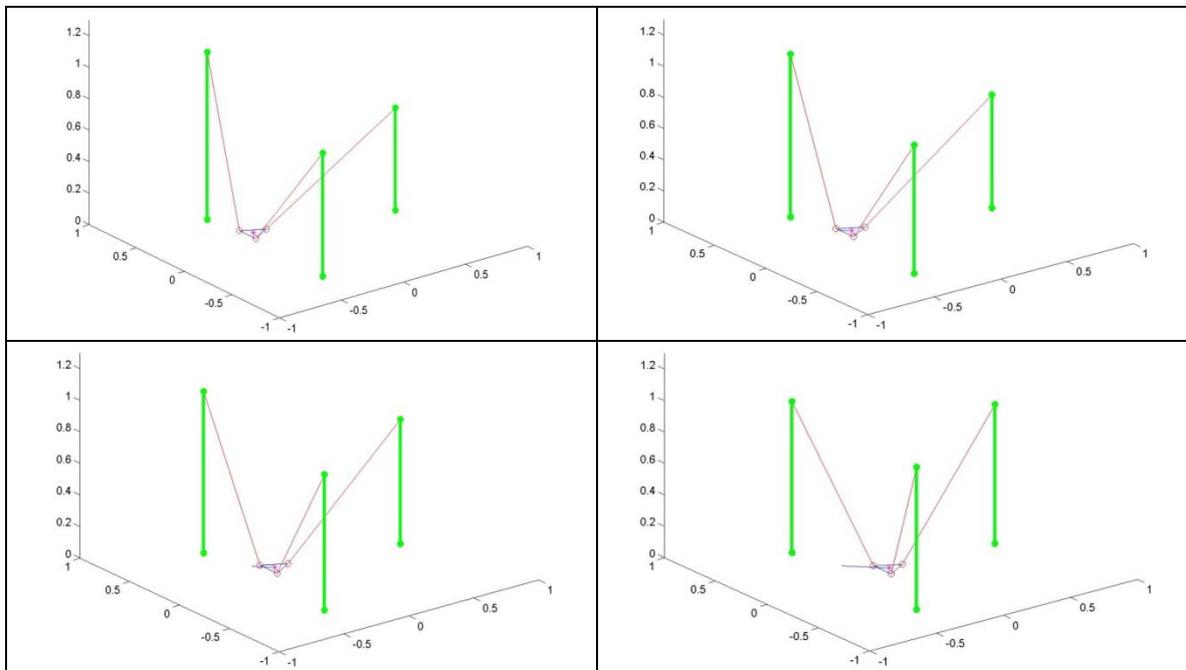


Figure V- 8 : Déplacements, Vitesses et Accélérations des moteurs (1<sup>ère</sup> cas Matlab)

**d) Cas  $n^{\bullet}=2$**

$$x = 0.0192 * t^2 - 0.0106 * t - 0.2447;$$

$$y = -x;$$



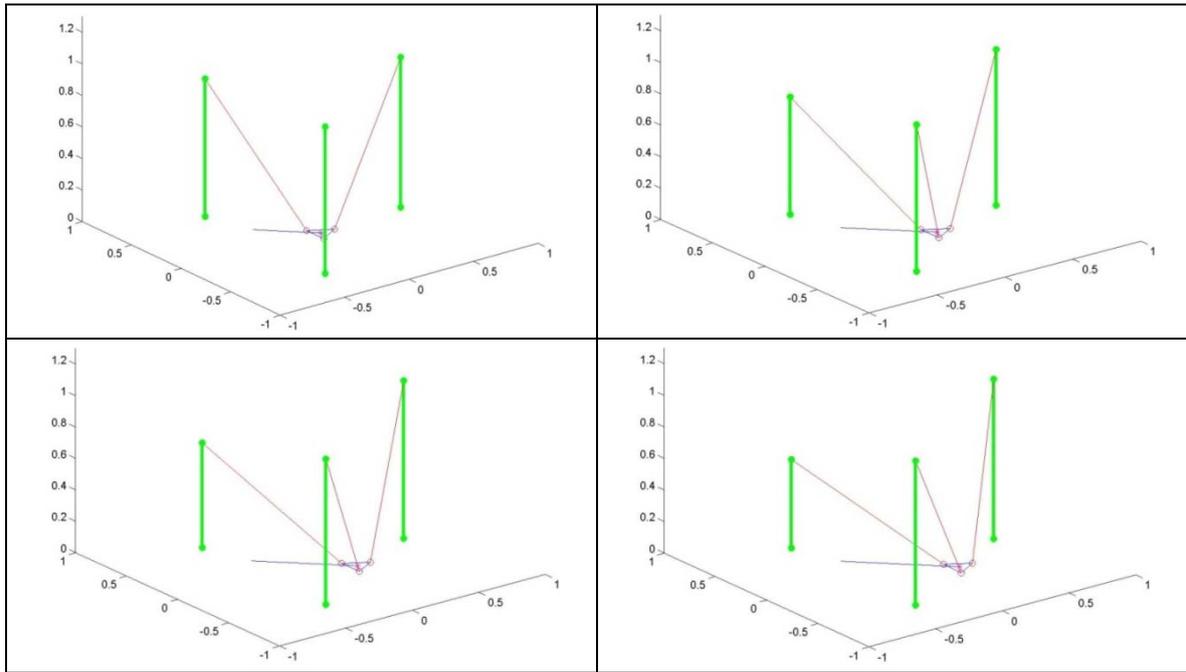


Figure V- 9 : Des instants de 2<sup>ème</sup> cas

➤ **Les Entrés :**

**1. Trajectoire de centre de la nacelle**

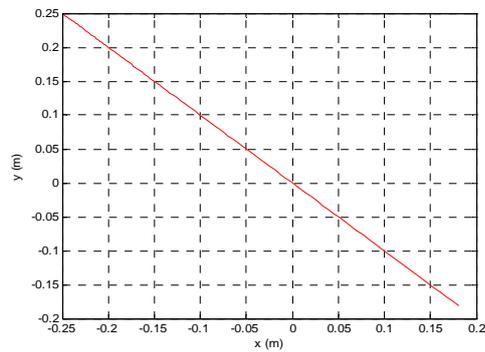


Figure V- 10 : Trajectoire de centre de la nacelle (2<sup>ème</sup> cas)

**2. Vitesse de centre de la nacelle**

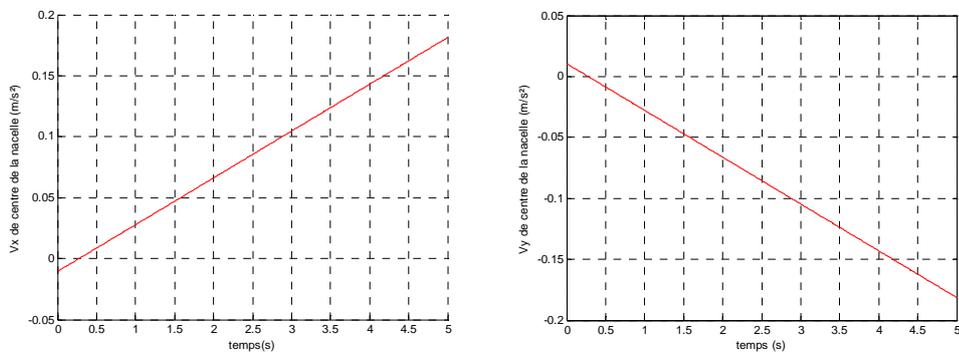


Figure V- 11 : Vitesse de centre de la nacelle (2<sup>ème</sup> cas)

3. Accélération de centre de la nacelle

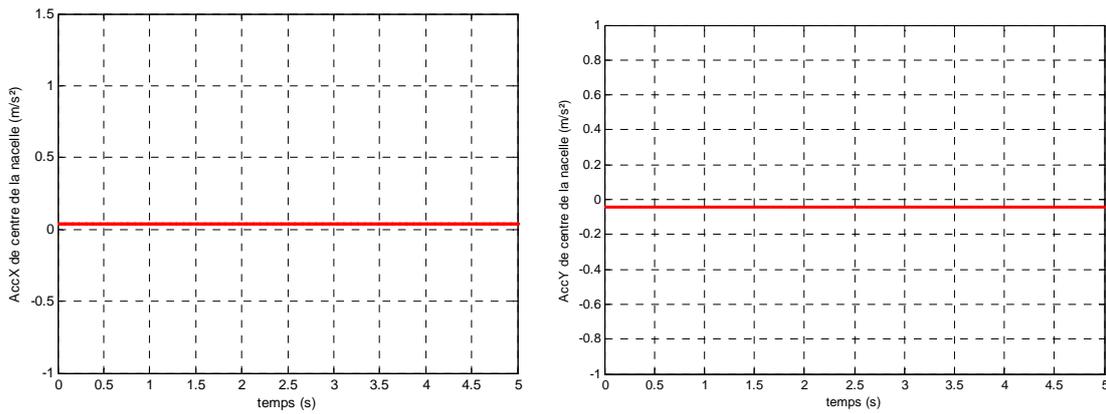


Figure V- 12 : Accélération de centre de la nacelle (2<sup>ème</sup> cas)

➤ Les Sortie

	Déplacement	Vitesse	Accélération
1 <sup>ère</sup> moteur			
2 <sup>ème</sup> moteur			
3 <sup>ème</sup> moteur			

Figure V- 13 : Déplacements, Vitesses et Accélérations des moteurs (2<sup>ème</sup> cas Matlab)

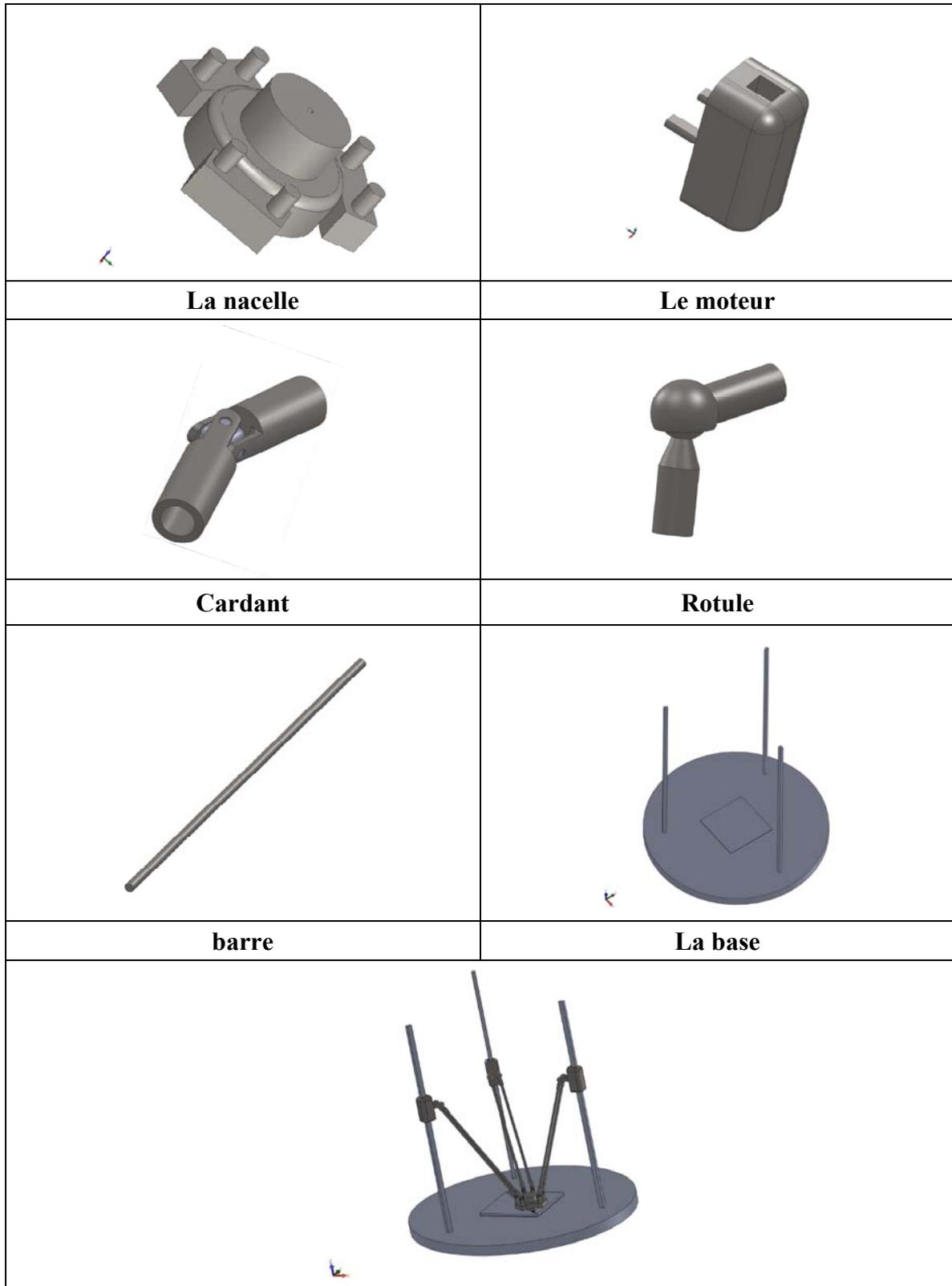
**V.3.3. Simulation par SolidWorks (COSMOS Work) :****V.3.3.1. Présentation des pièces de robot sous SolidWorks**

Figure V- 14 : les différentes pièces de robot Delta sous SolidWorks

### V.3.3.2. Les résultats par COSMOS Works :

Nous posons l'outil sur une trajectoire donnée. D'après une application de force sur l'outil qui va déplacer avec une vitesse et accélération sur cette trajectoire.

La trajectoire que nous avons donnée est semblable de la trajectoire de 2<sup>ème</sup> cas de la simulation sous MATLAB.

#### ➤ La Nacelle

##### 1. Trajectoire de centre de la nacelle

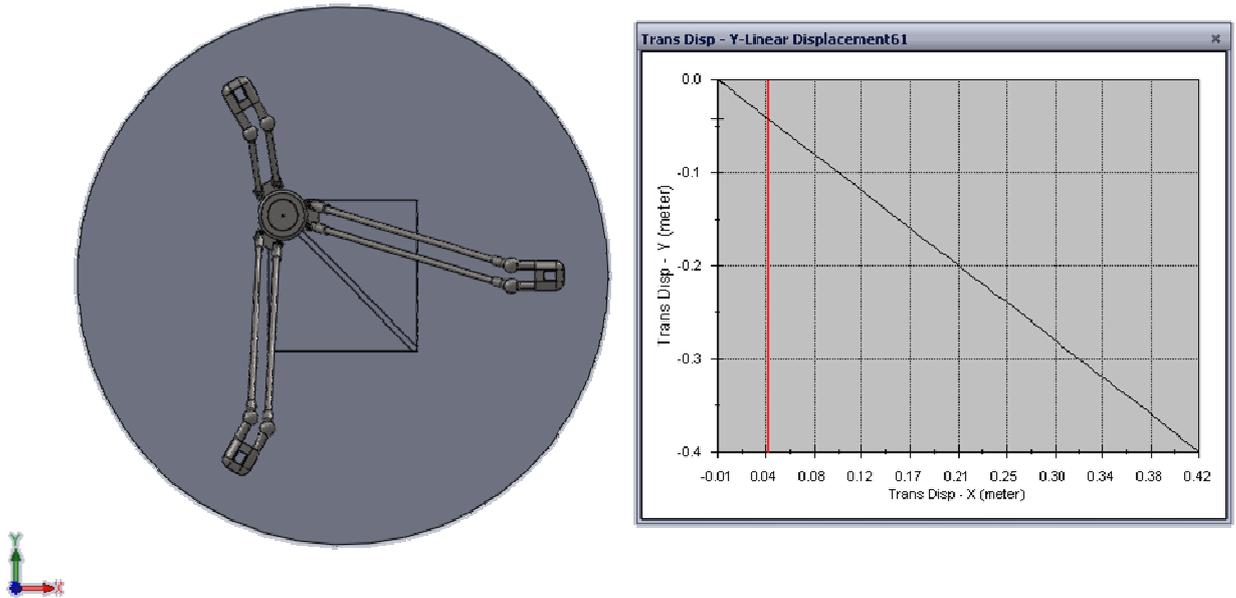


Figure V- 15 : Trajectoire de centre de la nacelle (COSMOS Works)

##### 2. Vitesse de centre de la nacelle

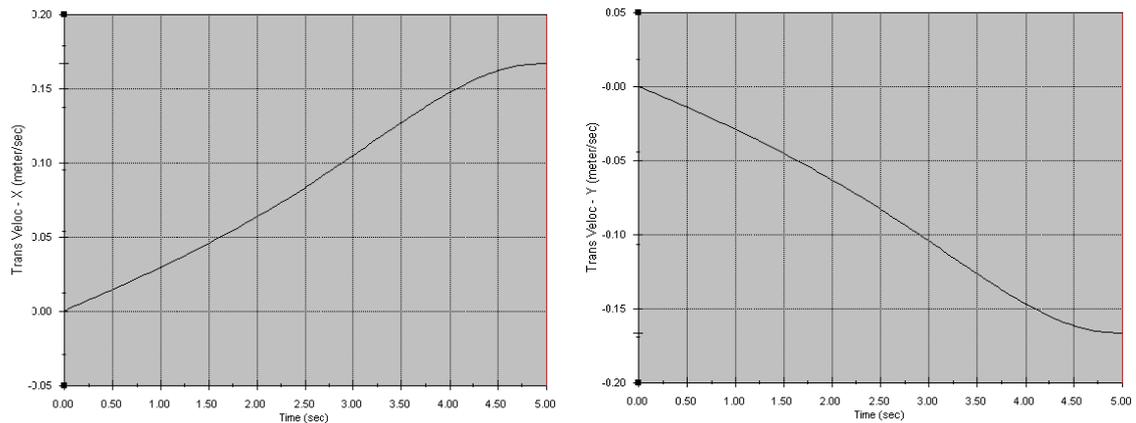


Figure V- 16 : Vitesse de centre de la nacelle (COSMOS Works)

### 3. Accélération de centre de la nacelle

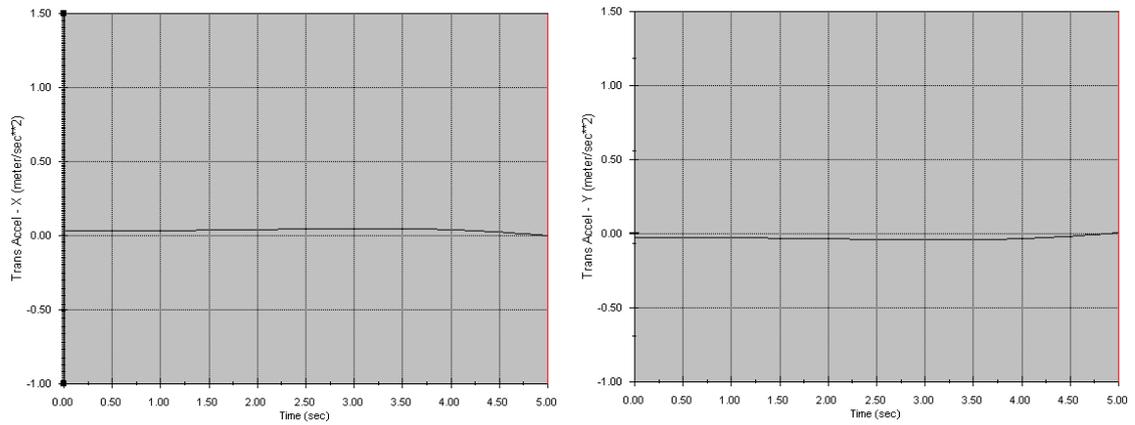


Figure V- 17 : Accélération de centre de la nacelle (COSMOS Works)

#### ➤ Les Moteurs

##### 1. Les déplacements des moteurs

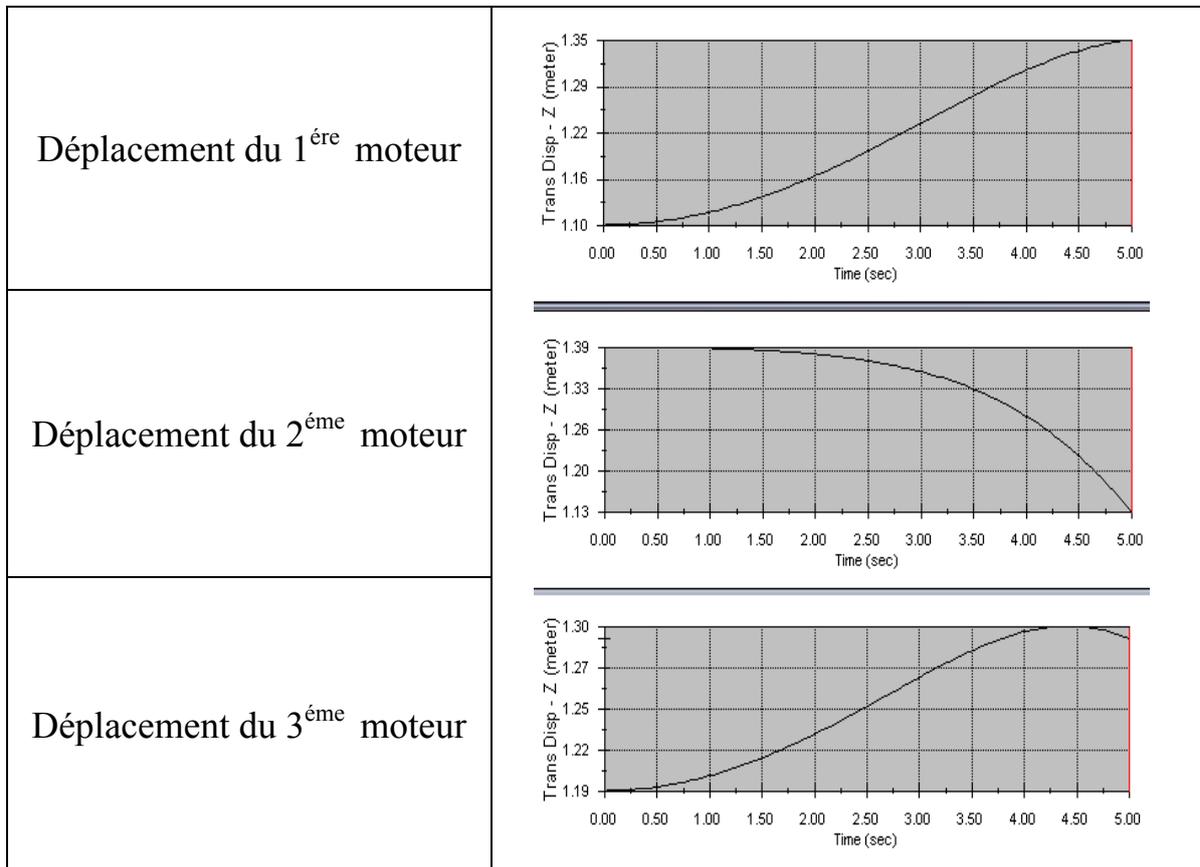


Figure V- 18 : les déplacements des moteurs (COSMOS Works)

2. Les vitesses des moteurs

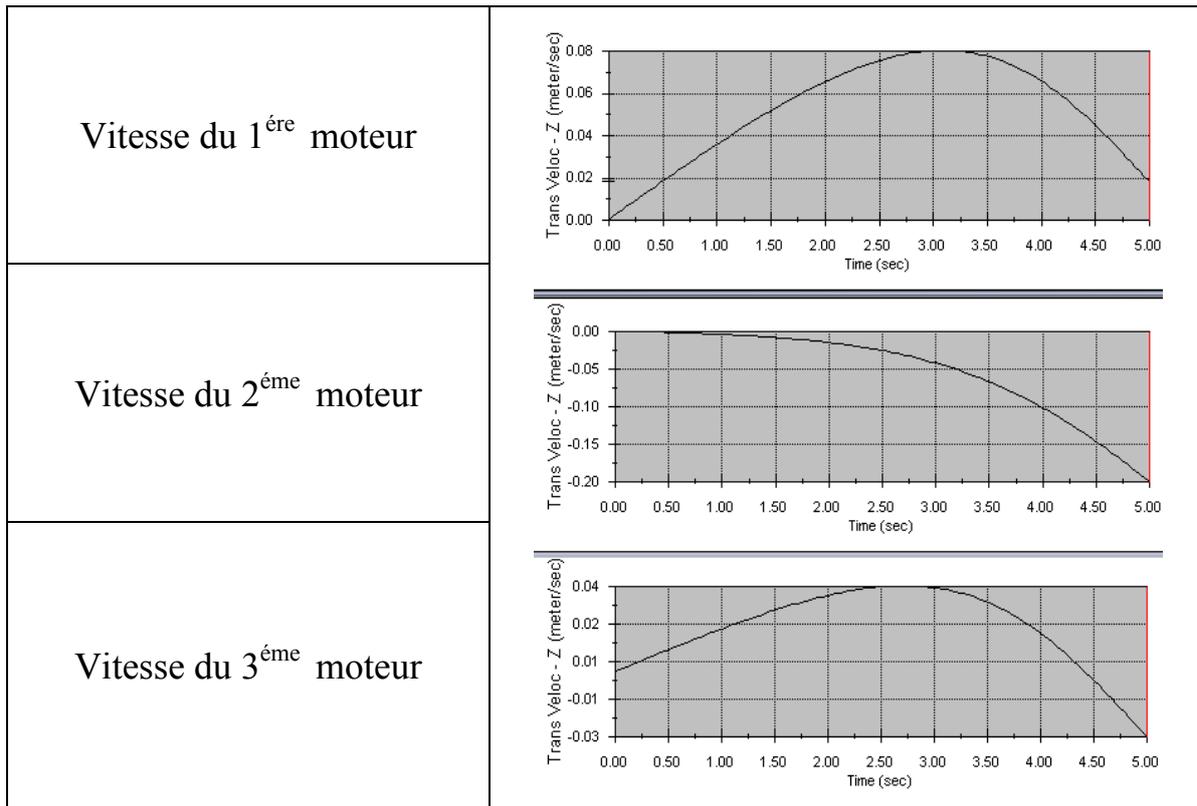


Figure V- 19 : les vitesses des moteurs (COSMOS Works)

3. Les accélérations des moteurs

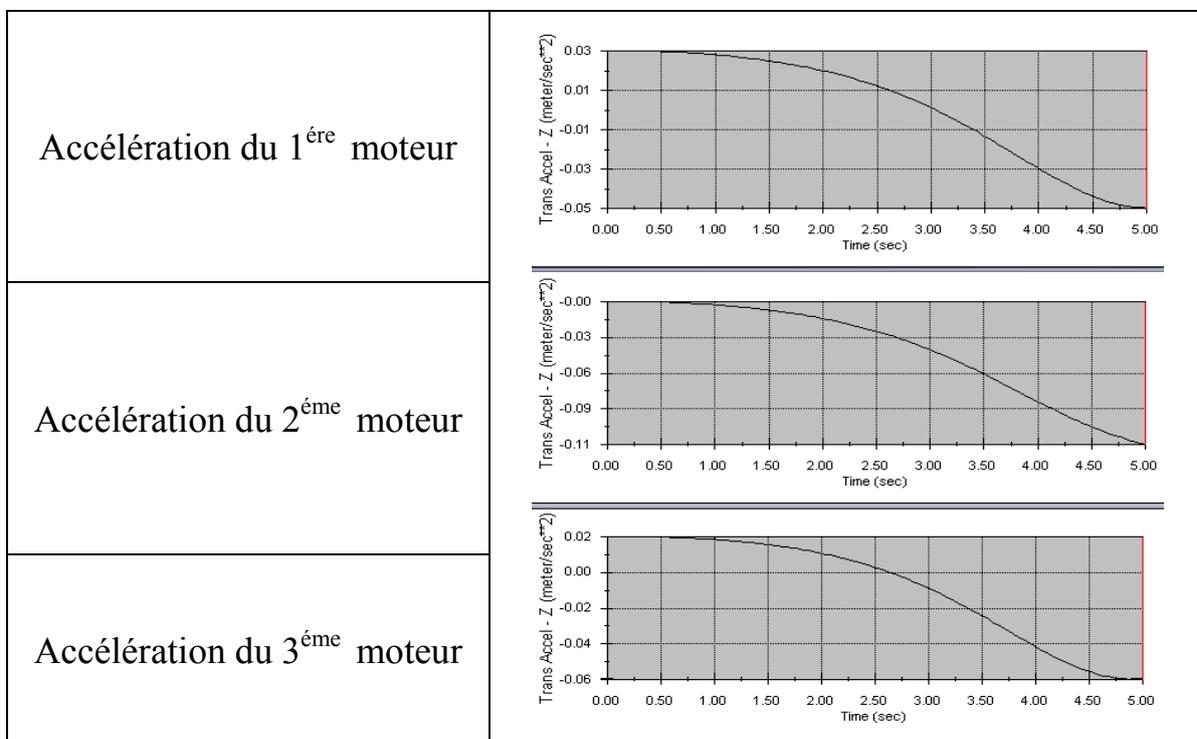


Figure V- 20 : les accélérations des moteurs (COSMOS Works)

# Conclusion générale

---

Dans ce mémoire, nous avons essayé de résoudre le problème d'optimisation multicritère de machine outil en architecture parallèle par l'utilisation d'une méthode capable pour ce type des problèmes qui est l'algorithme génétique NSGA II.

Le choix de l'architecture parallèle adéquate pour une machine-outil se fait après de comparaisons entre les caractéristiques de la machine-outil et les robots parallèles, tels que la performance dynamique, la précision et la rigidité. D'après cela nous avons trouvé seulement quelques architectures parallèles sont candidat en tant qu'architectures de machines-outils. Parmi ces architectures nous prenons le robot d'architecture Delta. En suite, nous avons choisi l'agencement de notre machine, afin de faire une modélisation géométrique, cinématique et dynamique de robot. D'après de sélectionner le type de topologie de la machine, il reste à effectuer une optimisation de ses dimensions.

Dans ce travail, nous avons formé un problème d'optimisation multi-objectif de notre machine car il est plus réel qu'une optimisation mono-objectif. Parmi les problèmes réels, on trouve les problèmes d'optimisation multi-objectifs qui ont démontré leur difficulté d'être résolus, les chercheurs ont appliqué plusieurs types de méthodes. Nous avons utilisé dans ce mémoire la technique de l'algorithme génétique (NSGA II) qui est une approche de résolution au sens de Pareto optimale, que nous avons implémenté sous Matlab. Nous avons choisi deux objectifs à optimisé avec six variable sous des contraintes. Les résultats obtenus montrent qu'il y a une influence de la longueur des barres de robot sur les deux objectifs qui sont l'accélération de la nacelle et le conditionnement de la matrice Jacobienne. En fin nous avons trouvé une solution optimale pour les dimensions de robot.

La qualité de résultat montre que l'algorithme génétique est capable pour résoudre des problèmes d'optimisation multi-objectifs complexe avec grand nombres des variables sous des contraintes compliqués avec un temps de calcul faible. Les performances du modèle proposé demeurent en effet perfectibles en améliorant notamment la précision et la rapidité de l'algorithme génétique lui-même.

Finalement, nous concluons par les perspectives suivantes : de généraliser cette application à tout genre de problèmes surtout les problèmes d'optimisation qui ont plus de deux objectifs.

## Références Bibliographiques

---

- [1] Merlet J-P., “ Les Robots parallèles “, Hermès, Paris, 1997.
- [2] Merlet J-P. , “Articulated device for use in particular in robotics”, United States Patent n° 5,053,687, 1 October 1991.
- [3] Merlet J-P., Perng MW., and Daney D., “*Optimal trajectory planning of a 5-axis machine tool based on a 6-axis parallel manipulator*”, In ARK , Piran, pages 315-322, 2000.
- [4] Merlet J-P, “Micro parallel robot mips for medical applications”, In IEEE Int. Conf. on Emerging Technologies and Factory Automation , Antibes, 2001.
- [5] Merlet J-P, “*Singular configurations of parallel manipulators and Grassmann geometry*”, Int. Journal of Robotics Research, 1989.
- [6] Merlet J-P, “*A parser for the interval evaluation of analytical functions and its applications to engineering problems*”, Journal Symbolic Computation, 2001.
- [7] Gough V.E., “Contribution to discussion of papers on research in automobile stability, control and tyre performance”, Proc. Auto Div. Inst. Mech. Eng., 1956-1957.
- [8] Stewart D., “A platform with 6 degrees of freedom”, Proc. Of the Institution of mechanical engineers, P. 371-386, 1965.
- [9] Clavel R., “DELTA, a fast robot with parallel geometry”, 18th Int. Symp. On Industrial Robot, Lausanne, 26-28 Avril 1988.
- [10] Sheldon P.C. “Six axis machine tool”, United States Patent n° 5,388,935 Giddings & Lewis, 14 Février 1995.
- [11] Pierrot F., Company O., “Towards non-hexapod mechanisms for high performance parallel machines”, Accepté à IECON 2000, Nagoya, 22-28 octobre 2000.
- [12] Pierrot F., Dauchez P., and Fournier A, “Fast parallel robots”, Journal of Robotic Systems, 1991.
- [13] Mourrain B., “The 40 generic positions of a parallel robot”, In Bronstein M., editor, ISSAC'93, ACM press, pages 173-182, Kiev (Ukraine), 1993.
- [14] Codourey A., “Control algorithm and controller for the direct drive Delta robot”, In 3rd IFAC/IFIP/IMACS Symp. Syroco'91, Vienne, 1991.
- [15] Yang G., Chen I-M., and Yeo S-H., “Design consideration and kinematic modeling for modular reconfigurable parallel robots”, In 10th World Congress on the Theory of Machines and Mechanisms, pages 1079-1084, Oulu, 1999.
- [16] Glazunov V-A and others, “*Classification principles and analysis methods for parallel-structure spatial mechanisms*”, Journal of Machinery Manufacture and Reliability, 1990.
- [17] Haugh E-J, Adkins F-A, and Luh C-M, “*Operational envelopes for working bodies of mechanisms and manipulators*”, ASME Journal of Mechanical Design, 1998.
- [18] Dietmaier P, “*The Stewart-Gough platform of general geometry can have 40 real postures*”, In ARK, pages 7-16, Strobl, 1998.

## Références Bibliographiques

---

- [19] Sergio L-A., « Optimisation multiobjectif et analyse de sensibilité appliquées à la conception de dispositifs », thèse de doctorat, l'école centrale de Lyon, 2006.
- [20] Alain B., « Optimisation multiobjectif et stratégies d'évolution en environnement dynamique », thèse de doctorat, l'Université des Sciences Sociales Toulouse I, 2001.
- [21] Srivas N. and Deb K., "Multiobjective Optimization using Nondominated Sorting in Genetic Algorithms", Technical Report, Department of Mechanical Engineering, Institute of Technology, India, 1993.
- [22] David S., "Multiple Objective Optimisation with Vector Evaluated Genetic Algorithm", In genetic Algorithm and their Applications : Proceedings of the First International Conference on Genetic Algorithm, p. 93-100, 1985.
- [23] Robin A., "Genetic Algorithm with Gender for Multi-Function Optimisation", TR. EPCC-SS92-01, Edinburgh Parallel Computing Center, Edinburgh, Scotland, 1992.
- [24] Michael P. Fourman, "Compaction of Symbolic Layout using Genetic Algorithms". In Genetic Algorithms and their Applications: Proceedings of the First International Conference on Genetic Algorithm, p. 141-153, 1985.
- [25] Mavrodīs C., "*Completely specified displacements of a rigid body and their application in the direct kinematics of in-parallel mechanisms*", Journal of mechanical design, décembre 99, vol 121, p. 485-491
- [26] Fonseca C-M. and Fleming P-J., "Genetic Algorithm for Multiobjective Optimization: Formulation, Discussion and Generalization," In: 5th Conference on Genetic Algorithm, San Mateo, California, pp. 416-423, Aug. 1993.
- [27] David S., "Multiple objective optimization with vector evaluated genetic algorithms," In: First International Conference on Genetic Algorithm, Lawrence Erlbaum, New Jersey, pp. 93-100, 1985.
- [28] Viennet C-F. and Marc I., "*Multicriteria Optimization using a Genetic Algorithm for Determining a Pareto Set*," International Journal of Systems Science, v. 27, n. 2, pp. 255-260, 1996.
- [29] Eckart Z. and Lothar T., "An evolutionary algorithm for multi-objective optimization: The strength Pareto approach". Technical report, Swiss Federal Institute of Technology Zurich, Zurich, 1998.
- [30] Eckart Z., "Evolutionary Algorithms for Multi-objective Optimization: Methods and Applications", Thesis Doctor of Technical Sciences, Swiss Federal Institute of Technology Zurich, 1999.
- [31] Eckart Z., Laumanns M. and Lothar T., "SPEA2: Improving the strength Pareto evolutionary algorithm," In: EUROGEN 2001, Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems, Athens, pp. 12-21, Sept. 2001.
- [32] Dorigo M. and Stützle T., "Ant Colony Optimization", MIT Press, 2004.
- [33] Chedmail P., Dombre E., Wenger P., « La CAO en robotique », Editions Hermès - 1998
- [34] Félix M. et al., "The Design Of Parallel Kinematic Machine Tools Using Kinetostatic Performance Criteria", UMR n°6597 CNRS, Institut de Recherche en Communications et Cybernétique de Nantes, France, 2007.
- [35] Oliver C., « Machines-outils rapides à structure parallèle Méthodologie de conception applications et nouveaux concepts », thèse de doctorat, Université Montpellier II, 2000.

## Références Bibliographiques

---

- [36] Clavel R., « Conception D'un Robot Prallele Rabide A 4 Degres De Liberte », thèse de doctorat, EPF de Lausanne 1991.
- [37] Luis Á-Silva, « Control Visual De Robots Paralelos. Analisis, Desarrollo Y Aplicacion A La Plataforma Robotenis », thèse de doctorat, Escuela Técnica Superior de Ingenieros Industriales, 2005.
- [38] John H., "*Outline for a logical theory of adaptive systems*", Journal of the Association of Computing Machinery, 3, 1962.
- [39] Goldberg D.E. "Genetic Algorithms in Search, Optimization and Machine Learning", Reading MA Addison Wesley, 1989.
- [40] Nicolas J., « Modélisation et résolution approchée de problème de tournées multi-objectif », thèse de doctorat, Université des Sciences et Technologies de Lille, 2004.

# Annexes

---

## A. Conventions de représentation :

La représentation sur papier des robots parallèles (et en général des structures spatiales articulées) est un exercice difficile. Il existe plusieurs types de représentation. Chacun d'eux permet une lisibilité accrue de certains paramètres, mais s'accompagne d'une perte d'information sur les autres. Parmi les types de représentation.

### A.1. Les vues en perspective (ou photographies)

Dans ce type de représentation, les articulations composant les chaînes cinématiques ne sont pas facilement visibles et identifiables (Figure A-1). Nous pouvons voir l'aspect général de la machine, mais pas de manière précise la disposition des articulations. Ce type de représentation n'est donc pas suffisant.



*Figure A- 20 : Photographie du robot Delta*

### A.2. Le dessin d'ensemble

Le dessin d'ensemble est utilisé en mécanique. Il sert à définir un mécanisme, son assemblage et son fonctionnement. Plusieurs vues extérieures, coupes et sections sont rassemblées sur un document. Ce type de représentation est particulièrement bien adapté pour les mécanismes plans, mécanismes qui sont couramment utilisés en mécanique. Par contre, l'utilisation de dessins d'ensemble devient réhhibitoire pour les mécanismes spatiaux complexes tels que les robots parallèles.

### A.3. Le schéma cinématique

## Annexes

La norme NF EN ISO 3952-1 codifie la représentation des schémas cinématiques (Figure A-2). Ces schémas, très pratiques pour représenter l'agencement des différentes liaisons composant un mécanisme, sont de lecture aisée pour les mécanismes plans et les mécanismes spatiaux simples, mais deviennent vite illisibles pour les mécanismes spatiaux complexes (Figure A-3).

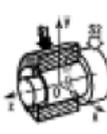
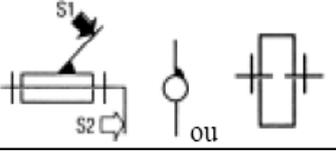
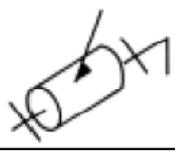
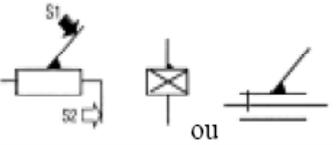
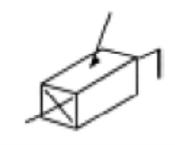
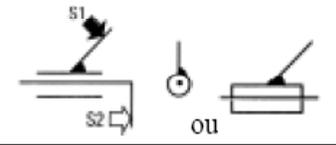
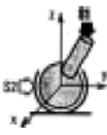
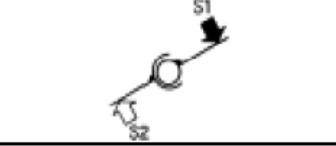
Liaisons	Exemples	Mouvements relatifs		Projection orthogonale 2D	Perspective 3D
		R	T		
Pivot		0	0		
		0	0		
		Rz	0		
Glissière		0	Tx		
		0	0		
		0	0		
Pivot glissant		0	0		
		0	0		
		Rz	Tz		
Rotule ou sphérique		Rx	0		
		Ry	0		
		Rz	0		

Figure A- 21 : Représentation normalisée de quelques liaisons.

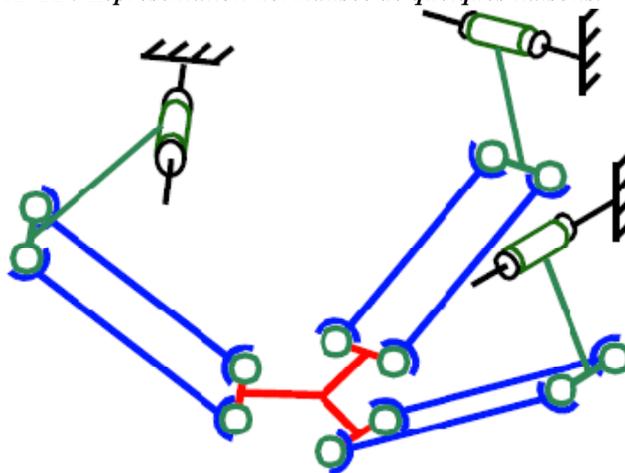


Figure A- 22 : Schéma cinématique du robot Delta.

### A.4. Les graphes d'agencement

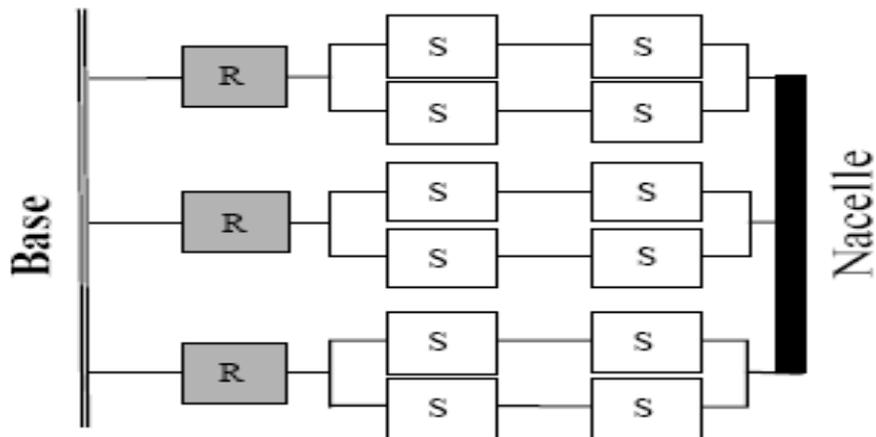
## Annexes

Ces graphes ont été proposés par François PIERROT dans. Les conventions retenues sont présentées (Figure A-4).

Nom de la liaison	Représentation	
	Liaison passive	Liaison motorisée
Rotoïde (pivot)	<b>R</b>	<b>R</b>
Prismatique (glissière)	<b>P</b>	<b>P</b>
Universelle (cardan)	<b>U</b>	
Sphérique (rotule)	<b>S</b>	

*Figure A- 23 : Conventions des graphes d'agencement.*

Avec ce type de représentation (Figure A-5), les informations concernant l'agencement géométrique des liaisons sont perdues. Par contre la comparaison des familles de mécanismes parallèles ainsi que le décompte des degrés de liberté sont facilités à l'aide de la formule de Grübler (A-1).



*Figure A- 24 : Graphe d'agencement.*

Par exemple, sur la Figure A-5, nous constatons que le robot se décompose en trois chaînes possédant les mêmes articulations, disposées en parallèle entre la base et la nacelle. Par contre, nous ne pouvons pas affirmer que ces trois chaînes sont identiques du point de vue de leur géométrie. Le nombre de degrés de liberté de ce mécanisme se calcule en utilisant la formule de Grübler (A-1) qui donne la mobilité d'un mécanisme dans le cas général, en dehors des positions et des agencements singuliers.

$$m = 6 N_p - 6 N_l + \sum_{i=1}^{N_l} dof_i - m_{int} \quad (A-1)$$

Où  $m$  est le nombre de degrés de liberté du mécanisme (en dehors configurations singulières),  $N_p$  le nombre de solides indépendants (bâti exclu),  $N_l$  le nombre de liaisons entre

ces solides,  $dof_i$  le nombre de degrés de liberté de la liaison numéro  $i$  et  $m_{int}$  est le nombre de mobilités internes.

Pour le robot Delta de la Figure I-5, nous avons  $N_p=10, N_l = 15, \sum_{i=1}^{N_l} dof_i = 39$   
 $m_{int} = 6$  Car chacune des barres à la possibilité de tourner sur elle-même (axe passant par le centre des deux liaisons rotule) sans que la position de la nacelle ne change. Finalement ; nous obtenons  $m = 6 \cdot 10 - 6 \cdot 15 + 39 - 6 = 3$ . Le mécanisme que nous avons représenté dans le graphe d'agencement de la Figure I-5 possède donc trois degré de liberté utiles. Par contre, ce type de calcul ne nous permet pas de connaître leur nature.

## B. Statut et résultat affichée pendant l'optimisation

Le GEATbx produit des rapports étendus de statut et de résultat pendant une optimisation. Par le défaut texte, l'information est écrite à la fenêtre de commande de Matlab (l'information de

statut). Cette information peut être sauvée en plus dans un fichier de texte pour la récupération et l'analyse postérieures (par défaut, cette information à ne pas sauvée).

### **1. Affichage de toutes les options utilisées au début d'une optimisation (en-tête d'information de statut)**

```
Evolutionary Optimization

Objective function: ObjFIN    Date: 08-Feb-2010    Time: 09:22:31
number of variables: 5
boundaries of variables:
    -0.25  -0.25   0.5   0.1   30
    0.25   0.25   0.7   0.3   50

Evolutionary algorithm parameters:
subpopulations = 8    individuals = 25 (at start per subpopulation)
termination 1: max. generations = 2000;
variable format = 0 (real values - phenotype == genotype)
selection
  function = selsus
  pressure = 1.9
  multiobj = 15
  gen. gap = 0.9
reinsertion
  rate = 1
  method = 2
recombination
  name = recdis
  rate = 0.6
mutation
  name = mutreal
  rate = 0.2
  range = 0.1 0.03 0.01 0.003 0.001 0.0003 0.1 0.03
precision = 12
regional model
migration
  rate = 0.1 interval = 20 topology = 0 selection = 1
output
results on screen every 10 generations
graphical display of results every 10 generations
method = 211111100
style = 111413212
results into text file every 10 generations
file name = res_gea.txt
```

### **2. L'information de statut montrée aux commandes fenêtre pendant l'optimisation (quelques lignes enlevées)**

Pendant l'optimisation l'information de statut est montrée dans le format tabulaire, une ligne par génération (ou aussi souvent que défini dans *Output.TextInterval*). L'en-tête de table fournit une description courte d'information contenue dans la table de statut. L'information de courir-statut contient:

- ✓ nombre de génération courante,
- ✓ nombre des fonctions objectives appelées,
- ✓ la meilleure valeur objective trouvée jusqu'ici (une valeur pour l'optimisation simple-objective, valeur multiple pour le cas multi-objectif),
- ✓ variables du meilleur individu (non affiché sur l'écran par défaut - peut être alimenté aussi bien),

## Annexes

- ✓ position des sous-populations (s'applique seulement en utilisant les sous-populations multiples),
- ✓ taille des sous-populations (s'applique seulement en utilisant la concurrence entre les sous-populations),
- ✓ l'information sur les critères d'arrêt (combien chacun des critères utilisés d'arrêt plein-est rempli),
- ✓ temps de calcul (heure pour une génération, heure globale pour l'optimisation jusqu'ici).

Les 5 derniers types d'information peuvent être commutés (On ou Off).

Generation	f-Count	Obj. Function	2. objv	3. objv	4. objv	Variables of best individual	Pos of subpopulations / size	Term: 1	Time: cpu/gen, full
1.	200	-77.474	2.6148	0	0	0.1829	0.079495 2 6 3 8 7 5 4 1	[ 0.05%]	( 0.00min 00:00:00)
20.	3544	-96.72	2.4462	0	0	0.11254	0.1925 2 3 5 7 6 8 4 1	[ 1.00%]	( 0.01min 00:00:04)
40.	7064	-102.51	2.5534	0	0	0.25	0.15117 4 2 7 8 1 5 6 3	[ 2.00%]	( 0.01min 00:00:10)
60.	10584	-100.25	2.4772	0	0	0.25	0.11936 5 1 2 7 3 4 6 8	[ 3.00%]	( 0.01min 00:00:15)
80.	14104	-98.043	2.4518	0	0	0.22709	0.12163 7 3 4 6 1 2 8 5	[ 4.00%]	( 0.01min 00:00:19)
100.	17624	-95.913	2.4048	0	0	0.033922	0.21318 8 6 4 3 2 1 5 7	[ 5.00%]	( 0.01min 00:00:24)
120.	21144	-111.15	2.6733	0	0	0.24324	0.21043 7 5 3 1 2 4 8 6	[ 6.00%]	( 0.01min 00:00:29)
140.	24664	-103.28	2.5386	0	0	0.16426	0.21228 7 4 5 1 2 3 6 8	[ 7.00%]	( 0.01min 00:00:34)
160.	28184	-101.56	2.5347	0	0	0.23828	0.15188 7 6 5 2 1 3 8 4	[ 8.00%]	( 0.01min 00:00:39)
180.	31704	-106.52	2.6149	0	0	0.21391	0.20911 7 6 4 2 1 3 8 5	[ 9.00%]	( 0.01min 00:00:45)
200.	35224	-105.32	2.5761	0	0	0.25	0.17082 5 6 8 1 2 3 4 7	[ 10.00%]	( 0.01min 00:00:50)

1880.	330904	-90.536	2.3088	0	0	0.10533	0.098588 7 5 2 4 1 3 6 8	[94.00%]	( 0.01min 00:09:18)
1900.	334424	-93.534	2.3713	0	0	0.033618	0.19054 8 6 3 2 1 5 4 7	[95.00%]	( 0.01min 00:09:25)
1920.	337944	-85.282	2.2267	0	0	-0.0061523	0.066466 8 6 4 2 1 3 5 7	[96.00%]	( 0.01min 00:09:31)
1940.	341464	-97.558	2.4308	0	0	0.063257	0.2151 6 5 4 3 1 2 7 8	[97.00%]	( 0.01min 00:09:37)
1960.	344984	-90.876	2.328	0	0	0.021362	0.16163 7 3 4 2 1 5 8 6	[98.00%]	( 0.01min 00:09:43)
1980.	348504	-100.15	2.477	0	0	0.17253	0.17454 7 5 2 4 1 6 8 3	[99.00%]	( 0.01min 00:09:50)
2000.	352024	-90.21	2.3003	0	0	0.084086	0.10505 8 5 2 3 1 4 7 6	[100.00%]	( 0.01min 00:09:57)

### 3. Résultat graphique pendant l'optimisation

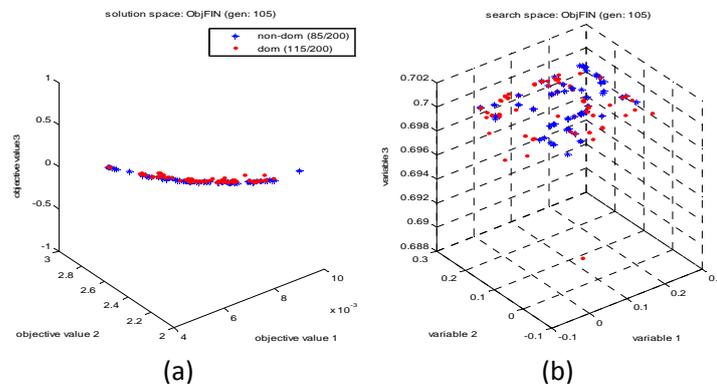
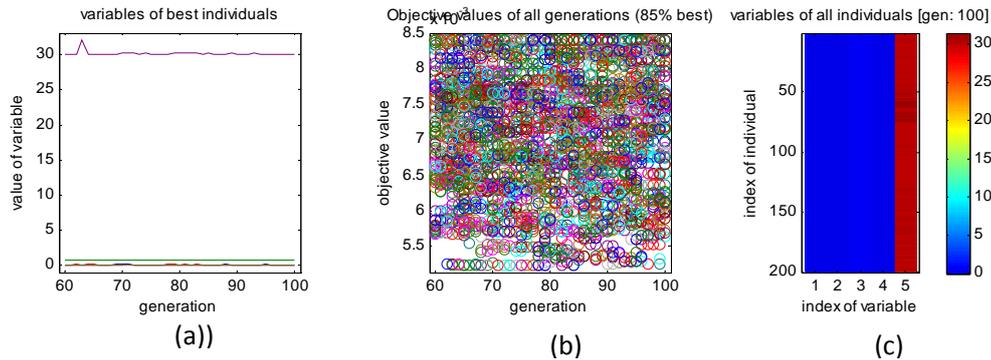


Figure B-1 : (a) Espace de solution. (b) Espace de recherche.



**Figure B- 2 :** (a) *Les valeurs de variables des meilleurs objectifs en fonction de générations.*  
 (b) *Les valeurs des objectifs en fonction de générations.*  
 (c) *L'évolution des variables individuels à chaque itération.*

#### 4. L'information de résultat montrée aux commandes fenêtre à la fin de l'optimisation

A l'fin du résultat d'optimisation (quand n'importe quel un des critères d'arrêt plein-est rempli) l'information est affichée sur l'écran (et économisé dans un dossier des textes s'indiqué). L'information de résultat contient:

- quels critères d'arrêt ont été utilisés (ici: nombre maximal des générations),
- combien de temps la course d'optimisation (nombre de générations et de temps),
- la meilleure valeur objective trouvée (une valeur pour la caisse simple-objective et valeurs multiples pour le cas multi-objectif) et quand ce résultat a été trouvé (nombre de génération),
- valeurs des variables du meilleur individu

End of optimization: max. generations (2000 generations; 9.98 cpu minutes / 9.98 time minutes)  
 Best Objective value: -90.554 2.7638 0 0 in Generation 720  
 Best Individual: 0.23753 0.2391 0.69981 0.10004 30  
 Non-dominated individuals: 200 (of 200 individuals in return population)

## C. Vérification de l'hypothèse des masses corrigées.

Un grand nombre de simulations réalisées à l'aide du logiciel de simulation dynamique ADAMS permet de vérifier la validité de cette approximation. Par exemple, en modélisant une machine dont les caractéristiques géométriques sont  $r \square 200mm$ ,  $R \square 800mm$ ,  $d \square 100mm$ ,  $L \square 781mm$  et dont les masses des éléments en mouvement sont  $M_1 \square 70kg$ ,  $M_2 \square 70kg$  et  $M_b \square 10 kg$ .

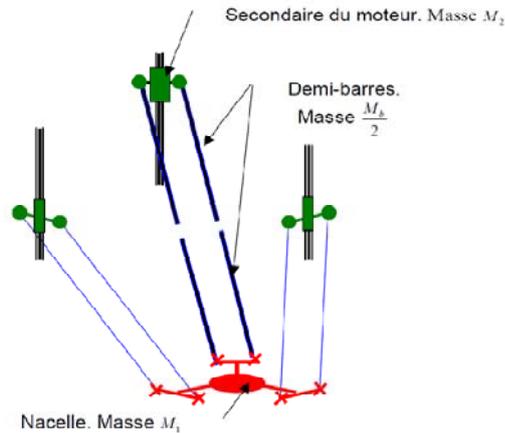


Figure C -1 : Répartition de la masse des barres.

Nous obtenons, en appliquant un effort moteur de 4000N aux deux premiers moteurs et de -8000N au troisième moteur, la norme de l'accélération résultante de la nacelle. Ensuite, nous comparons la norme de cette accélération dans trois cas (voir résultats Figure C-2) avec le logiciel ADAMS, en tenant compte de la masse des barres et en faisant l'hypothèse des masses corrigées et enfin, en intégrant l'équation différentielle du mouvement avec MATLAB.

Nous constatons que l'erreur entre les deux modèles ADAMS (barres avec masse et barres sans masse) est inférieure à 10%, ce qui est tout à fait acceptable pour les calculs de pré-dimensionnement que nous allons effectuer.

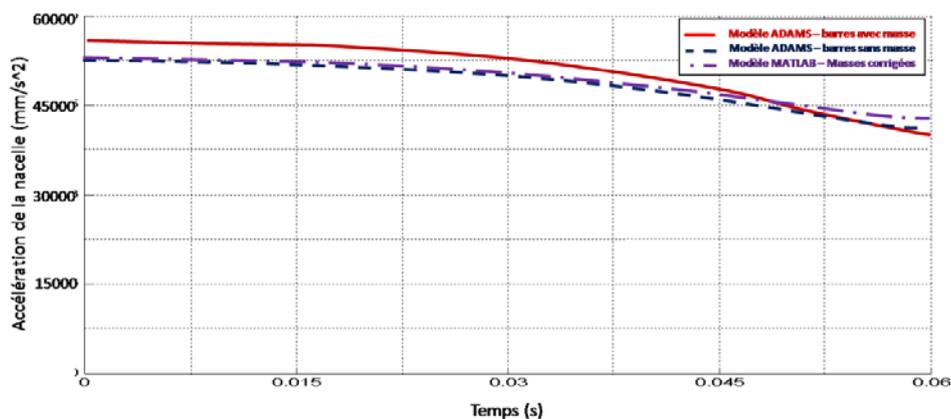


Figure C -2 : Vérification de l'hypothèse des masses corrigées [35]