



HAL
open science

Conception préliminaire de manipulateurs mobiles et génération de consignes évolutionnaires : une méthodologie pour intégrer la commande dans l'évaluation de la structure

Sophie Sakka

► **To cite this version:**

Sophie Sakka. Conception préliminaire de manipulateurs mobiles et génération de consignes évolutionnaires : une méthodologie pour intégrer la commande dans l'évaluation de la structure. Automatique / Robotique. Université Pierre et Marie Curie - Paris VI, 2002. Français. NNT : . tel-00120472

HAL Id: tel-00120472

<https://theses.hal.science/tel-00120472>

Submitted on 15 Dec 2006

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

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

THESE de DOCTORAT

de

L'UNIVERSITE PARIS 6

présentée par

Sophie SAKKA

pour obtenir le grade de

DOCTEUR DE L'UNIVERSITE PARIS 6

Spécialité : Robotique

**Conception préliminaire de manipulateurs mobiles et
génération de consignes évolutionnaires : une
méthodologie pour intégrer la commande dans
l'évaluation de la structure**

Soutenue le 20 septembre 2002

JURY

M. J. Pontnau	Professeur à l'Université de Créteil	Rapporteur
M. W. Khalil	Professeur à l'Ecole Centrale de Nantes	Rapporteur
M. R. Chellali	Maître-assistant à l'Ecole des Mines de Nantes	Co-rapporteur
M. P. Bidaud	Professeur à l'Université Pierre et Marie Curie	Examineur
M. J.-G. Fontaine	Professeur à l'ENSIB	Examineur
M. J.-F. Le Maitre	Professeur à l'Université Aix-Marseille	Examineur
M. N.K. M'Sirdi	Professeur à l'Université Versailles-S ^t -Quentin	Examineur
M. K. Youcef-Toumi	Professeur au M.I.T Cambridge (USA)	Examineur

Table des matières

Introduction Générale	1
1 Conception robotique optimisée	5
1.1 Conception optimisée	5
1.1.1 Formalisation du problème de conception	5
1.1.2 Les étapes de conception	6
1.1.3 Optimisation pour la conception	8
1.1.4 Conception robotique	8
1.2 Optimisation pour les systèmes robotiques	9
1.2.1 Problème de l'optimisation globale	9
1.2.2 Spécificités de l'optimisation	10
1.2.3 Méthodes d'optimisation	14
1.3 Le problème de la commande dans la conception préliminaire robotique	17
1.3.1 Problème général	17
1.3.2 Méthodologies existantes	18
1.3.3 Le comportement dans l'évaluation de la structure	19
1.4 Manipulateurs mobiles	19
1.4.1 Robots manipulateurs mobiles	20
1.5 Approche proposée	23
1.5.1 Conception préliminaire orientée tâche	23
1.5.2 Optimisation globale	24
1.5.3 Conclusion	26
2 Approche par l'Optimisation Evolutionnaire	29
2.1 De l'évolution à l'évolutionnaire	29
2.1.1 Origines	29
2.1.2 Philosophies évolutionnaires	31
2.2 Algorithme Génétique Simple (AGS)	32
2.2.1 Initialisation	32
2.2.2 Evaluation	33
2.2.3 Sélection	33
2.2.4 Croisement à un point	33

2.2.5	Mutation binaire	34
2.2.6	Remplacement générationnel	34
2.3	Nature et limites de la méthode	34
2.3.1	Diversité des algorithmes	34
2.3.2	Avantages et inconvénients	35
2.3.3	Convergence et théorème des schémas	36
2.3.4	Dilemme exploration/exploitation	37
2.3.5	Pré-requis pour l'optimisation évolutionnaire	38
2.4	Suivi de trajectoire	38
2.4.1	Position du problème	39
2.4.2	Algorithme évolutionnaire	41
2.4.3	Premiers résultat de simulation	51
2.5	Conclusions	60
2.5.1	Egarement de la recherche	60
2.5.2	Perte des bonnes solutions	61
2.5.3	Caractéristiques de l'évolution	61
2.5.4	Conclusion	61
3	Générateur de consignes évolutionnaire	63
3.1	Amélioration des performances	63
3.1.1	Adaptation du génotype	64
3.1.2	Fonction d'adaptation	65
3.1.3	Mémoire évolutionnaire	68
3.1.4	Dynamique et mémoire évolutionnaires	72
3.1.5	Conclusion	74
3.2	Cas de la manœuvre	75
3.2.1	Position du problème	75
3.2.2	Paramètres influants	76
3.2.3	Optimisations imbriquées	77
3.2.4	Tâche et robots	79
3.3	Algorithme évolutionnaire	79
3.3.1	Représentation des solutions	79
3.3.2	Evaluation des solutions	80
3.3.3	Opérateurs et paramètres d'évolution	83
3.3.4	Résultats de simulation	84
3.4	Chromosomes mixtes de longueurs variables	86
3.4.1	Intérêt	86
3.4.2	Construction initiale des chromosomes	87
3.4.3	Manipulation des chromosomes	88
3.4.4	Convergence rapide	89
3.5	Conclusion	90

4	Elimination préliminaire des structures	93
4.1	Position du problème	93
4.1.1	Objectif	93
4.1.2	Décomposition de la tâche	94
4.1.3	Limites de l'étude : paramètres fixés	95
4.1.4	Paramètres non valués	96
4.2	Conception préliminaire	96
4.2.1	Evolutions imbriquées	96
4.2.2	Cinématique modulaire	97
4.2.3	Modélisation et contraintes géométriques	99
4.2.4	Evaluation des solutions	101
4.2.5	Algorithme Evolutionnaire multi-chromosomique	104
4.3	Résultats de simulation	109
4.3.1	Tâche et sous-tâches	109
4.3.2	Comportement de l'algorithme	111
4.3.3	Solutions topologiques retenues	114
4.4	Conclusion	116
5	Optimisation commande/structure	119
5.1	Problématique générale	119
5.1.1	Avancement / Objectif	119
5.1.2	Bilan des données	120
5.2	Recherche des consignes : stabilité en complexité	121
5.3	Détermination d'une configuration initiale	123
5.3.1	problématiques de la recherche	123
5.3.2	Algorithme	124
5.3.3	Temps de simulation	127
5.3.4	Conclusions sur le générateur de consignes	131
5.4	Sélection finale	131
5.5	Conclusions sur la méthodologie générale	133
5.5.1	Optimisations successives	133
5.5.2	Performances de l'algorithme	135
5.5.3	Commentaires sur l'outils évolutionnaire	137
	Conclusion Générale	139
	Bibliographie	143
	Annexes	151
A	Etat de l'art en Manipulation Mobile	153
A.1	Robots manipulateurs mobiles	153
A.1.1	Bref recadrage historique	154
A.1.2	Actualités et perspectives pour les manipulateurs mobiles	155
A.1.3	Problématique de la manipulation mobile	156

A.2	Optimisation pour les manipulateurs mobiles	159
A.2.1	Optimisation de la structure	159
A.2.2	Notations et définitions générales	160
A.2.3	Solutions orientées tâche	161
A.2.4	Conclusion	165
B	Représentations mathématiques des chaînes cinématiques	167
B.1	Matrices de passage homogènes	167
B.2	Convention d'Euler	168
B.3	Paramétrage de Denavit-Hartenberg	168
C	Caractéristiques dynamiques de RoMAin et du PUMA 560	171
C.1	Modèle dynamique de ROMAIN	171
C.1.1	Equations de contraintes	171
C.1.2	Modèle dynamique pour la simulation	172
C.2	Paramètres du PUMA 560	173
D	Sources des opérateurs évolutionnaires en langage C	175
D.1	Initialisation	175
D.2	Croisement	176
D.2.1	Croisement à 1 point de césure	176
D.2.2	Croisement uniforme	176
D.2.3	Reproduction des individus	177
D.3	Mutation	177
D.3.1	Mutation binaire	177
D.3.2	Mutation réelle	177

Table des figures

1.1	Les différentes étapes de la conception.	7
1.2	Approche dissociée de la manipulation mobile.	22
1.3	Conception préliminaire orientée tâche d'un manipulateur mobile.	23
2.1	Généalogie Evolutionnaire.	30
2.2	Algorithme Génétique Simple (AGS).	32
2.3	Diversité des algorithmes évolutionnaires.	35
2.4	Représentation de la trajectoire et configuration initiale du manipulateur mobile	40
2.5	Representation du chromosome lié au vecteur des consignes	43
2.6	Evitement de collision avec le mur	47
2.7	Evolution de la fonction e^{-x} autour de 0.	48
2.8	Principe du croisement à deux points de césure	49
2.9	Algorithme pour la simulation	52
2.10	Evolution de la fonction d'adaptation au cours des générations (— Meilleur - - - Moyenne)	53
2.11	Evolution de la valeur des critères des meilleurs individus en fonction des générations	54
2.12	Comparaison des évolutions avec variation de la tolérance à l'écart à la trajectoire ε_ℓ (— Meilleur - - - Moyenne).	55
2.13	Deux trajectoires équivalentes du point de vue du critère d'erreur C_2	56
2.14	Evolution de la fonction d'évaluation en fonction des générations (— Meilleur - - - Moyenne)	57
2.15	Influence du paramètre de mutation p_m sur l'évolution (— Meilleur - - - Moyenne)	59
3.1	Exemple de surface d'écart à la trajectoire désirée à la fin d'une simulation	65
3.2	Evolution de la valeur d'adaptation des individus en fonction des générations après modification du critère d'erreur (— Meilleur - - - Moyenne)	66
3.3	Evolution de la valeur des critères pour les meilleurs individus en fonction des générations après modification du critère d'erreur	67
3.4	Graphes d'évolution utilisant l'opérateur de ré-introduction des meilleurs individus (— Meilleur - - - Moyenne)	69

3.5	Influence du paramètre de mémorisation sur l'évolution avec $p_m = 0.0125$	70
3.6	Statistiques de la valeur d'adaptation en fonction de p_r et de la génération	71
3.7	Influence du paramètre de mutation sur l'évolution avec $p_r = 10$ et $p_r = 3$ pour les meilleurs individus	72
3.8	Courbes d'évolution finales (— Meilleur - - - Moyenne)	73
3.9	Recherche d'une trajectoire "optimale" dans le cas d'un suivi de paroi .	75
3.10	Une répartition temporelle des vecteurs d'entrées de commande dans le cas $n = 4$	76
3.11	Génération d'une commande par optimisations imbriquées.	77
3.12	Algorithme général pour la détermination d'une commande évolutionnaire optimale.	78
3.13	Suivi de trajectoire avec franchissement de marche	79
3.14	Schématisation des critères pris en compte dans les optimisations inférieure et intermédiaire	81
3.15	Ellipsoïde d'évitement d'obstacles	82
3.16	Convergence de l'adaptation inférieure, $n = 4$ et $F = 1$ (— — — Meilleur individu — — — Moyenne de la population)	84
3.17	Convergence de l'adaptation intermédiaire, moyenne sur 100 évolutions pour $n = 4$ (— Meilleur individu - - - Moyenne de la population)	85
3.18	Convergence de l'adaptation supérieure, moyenne sur 100 évolutions (— Meilleur individu - - - Moyenne de la population)	86
3.19	Construction initiale des k individus supérieurs ($j = 1..k$).	87
3.20	Croisement de deux chromosomes parents a et b de longueurs différentes. Cas où $n_a < n_b$	88
3.21	Evolution de l'adaptation avec utilisation de chromosomes mixtes de longueur variable.	90
4.1	Extraction de quatre sous-tâches caractéristiques d'une tâche globale. .	94
4.2	Décomposition de la conception préliminaire.	97
4.3	Assemblage de 3 modules différents pour la constitution d'un robot. . .	98
4.4	Notations.	100
4.5	Génotype mixte dans le cas d'un robot à 3 ddl.	105
4.6	Synthèse de mécanismes pour l'élimination préliminaire.	108
4.7	La tâche et sa décomposition.	109
4.8	Adaptation globale et locale pour la synthèse par optimisations imbriquées.	112
4.9	Evaluations intermédiaires, critères inférieurs et globaux	113
4.10	Statistiques de convergence, sur 100 simulations, avec et sans la contrainte initiale de remplacement des topologies.	114
4.11	Cotes des extrémités inférieures des chaînes cinématiques.	116
4.12	Exemples de manipulateurs proposés.	117
5.1	Avancement de l'étude.	119
5.2	Evaluations intermédiaires, critères inférieurs et globaux	122

5.3	Algorithme de recherche d'une configuration initiale permettant une valeur optimale du premier vecteur de consignes.	124
5.4	Les deux évaluations locales pour quantifier la valeur d'une configuration initiale.	125
5.5	Configurations validées de la plate-forme mobile à la base du manipulateur	127
5.6	Algorithme pour l'utilisation des performances statistiques de la recherche des configurations initiales.	129
5.7	10 meilleures configurations de la plate-forme mobile, qui définissent les intervalles des configurations testées par U_1	130
5.8	Cheminement jusqu'à la solution finale.	132
A.1	Manipulateurs mobiles en fonction	155
A.2	Utilisation de la redondance pour déterminer une configuration confortable.	157
A.3	Paramètres de la base dans le référentiel fixe.	160
B.1	Convention de Denavit-Hartenberg.	169

Introduction Générale

L'intervention en milieux hostiles pour l'homme nécessite l'utilisation de systèmes fiables, performants et adaptés aux conditions. La conception de systèmes robotiques consiste en la mise en œuvre de machines autonomes qui répondent aux besoins spécifiques liés à ces missions. Les progrès de l'informatique et les demandes sans cesse croissantes dans ce domaine permettent la résolution de problèmes de plus en plus complexes ainsi que la formalisation du processus de conception, qui se décompose alors en plusieurs étapes [Yoshikawa80][Oueddou99]. La **conception préliminaire** est une étape centrale de la conception robotique qui consiste à extraire par la simulation une solution structurelle optimale du point de vue de critères d'optimisation [Chedmail98][Oueddou99][Regnier96]. Elle soulève des problèmes essentiels, liés à la formulation de la fonction d'évaluation (qui met souvent en œuvre des critères hétérogènes menant à son irrégularité), à la grande taille des espaces de recherche et à la complexité du système étudié. La résolution des problèmes peut alors se faire par une méthode spécifique au problème traité ou en utilisant une méthode adaptée aux grands espaces de recherche et aux fonctions d'évaluation irrégulières (méthodes naturelles).

L'évaluation des solutions reste cependant problématique dans le cas où la tâche à réaliser nécessite des déplacements. En effet, si la mise en place de critères évolués permet de déterminer des structures localement compétentes, l'influence des déplacements du robot lors de la réalisation de la tâche sur sa structure n'est que peu utilisée du fait de la difficulté de commander des systèmes dont les caractéristiques cinématiques et dynamiques ne sont pas fixées. L'introduction de la commande dans la boucle d'optimisation de la structure autoriserait alors une évaluation plus complète des solutions avant leur prototypage et les tests en conditions réelles.

Ce manuscrit propose une analyse de méthodologie de la conception préliminaire afin d'apporter des solutions à ces problèmes de synthèse parallèle de mécanismes et de consignes encore peu exploités. La famille robotique étudiée est celle des manipulateurs mobiles. Ces robots, constitués de l'assemblage d'une plate-forme mobile et d'un ou plusieurs bras manipulateurs suscitent un grand intérêt du fait de leurs capacités d'adaptation à de nombreuses tâches ainsi que d'autonomie en intervention. Si des solutions ont été proposées aux problèmes de commande inhérents à une telle asso-

ciation de mécanismes [Foulon98][Cellier96][Rey97][Yamamoto94(b)][Perrier97]..., leur conception structurelle en tant qu'entité robotique à part entière n'est pas encore d'actualité. Celle-ci dépend des travaux effectués, d'une part sur les plate-formes mobiles, d'autre part sur les chaînes cinématiques ouvertes. Nous proposons, dans ce manuscrit, de valider la méthodologie proposée sur la détermination des paramètres structuraux optimaux d'une chaîne cinématique ouverte qui permettra, fixée sur une plate-forme mobile donnée, de réaliser une tâche de manipulation mobile. L'interaction synthèse de mécanismes / synthèse de consignes est développée à travers la mise en place d'optimisations successives déterminant, d'une part, un ensemble de structures localement adaptées à la tâche, d'autre part, pour chacune de ces structures, les consignes qui permettent, lorsque cela est possible, de réaliser la tâche globale. La solution structurelle retenue sera ainsi optimale du point de vue de ses performances face à la réalisation de la tâche.

L'utilisation des algorithmes évolutionnaires est particulièrement bien adaptée pour aborder, d'une part l'optimisation de structures modulaires, d'autre part l'optimisation des consignes à appliquer. Les travaux utilisant cette méthode d'optimisation naturelle qui se sont intéressés à la synthèse préliminaire [Chocron00] ou à la génération de configurations ou de consignes [Chocron00][Leger99] ne sont malheureusement pas allés jusqu'à l'établissement d'une commande autorisant des retours d'informations sur la réalisation de la tâche. Dans l'approche que nous proposons, les différentes évaluations mises en place sont essentiellement orientées vers la tâche à réaliser. Celles-ci permettent alors une abstraction du système robotique étudié et donc de conférer un caractère générique à l'approche.

Ce document comporte cinq chapitres. Le premier présente les différents niveaux de la problématique et comporte une étude bibliographique sur les domaines concernés. Ceux-ci sont très variés et comprennent la définition de l'existant en conception préliminaire robotique ainsi que des problèmes qui lui sont liés. Le problème de la mise en place d'une commande "générique" y est abordé, ainsi qu'une description de la problématique de la manipulation mobile. Cette dernière est complétée d'un état de l'art plus précis fourni en annexe 1. La méthodologie générale de la conception préliminaire que nous proposons y est décrite. Sa validité repose entièrement sur l'établissement d'un générateur de consignes dont la mise en place est soumise à des contraintes de réalisation très fortes : fiabilité (trouver systématiquement une solution, lorsqu'elle existe), robustesse (s'adapter à la grande diversité des structures), rapidité (test de nombreuses solutions topologiques) et généralité (peu d'informations disponibles sur les systèmes étudiés). Les deux chapitres suivants lui sont donc dédiés.

Les deuxième et troisième chapitres aboutissent à la mise en place d'un générateur d'entrées de commande évolutionnaire sous les contraintes précédemment évoquées. Le chapitre 2 se concentre plus particulièrement sur le fonctionnement des algorithmes évolutionnaires, et propose dans un premier temps une mise en place classique du

problème de la commande. Il permet de mettre en évidence différents problèmes qui apparaissent lors de l'utilisation de cette méthode d'optimisation (liés à l'interprétation de la pose du problème d'optimisation et au système étudié) et d'en bien comprendre le fonctionnement. Le chapitre 3 met en œuvre les adaptations permettant parer aux défaillances mises en évidence au chapitre précédent. La synthèse des consignes, sur des robots connus et pour des tâches de complexité de réalisation croissante, est validée de manière fiable et systématique.

Basés sur cet outil de génération de consignes performant, les deux derniers chapitres décrivent la synthèse de mécanismes dont l'évaluation est orientée sur la réalisation de la tâche. La méthodologie divise cette étude en deux modules. Le quatrième chapitre effectue une élimination préliminaire des chaînes cinématiques inadaptées à la réalisation de la tâche. Cette étude de synthèse préliminaire de mécanismes se base sur une décomposition de la tâche en sous-tâches caractéristiques et ne prend pas en compte les mouvements du robot dans son évaluation. La détermination des structures retenues peut alors être effectuée grâce à des méthodes existantes [Chocron00], en les adaptant toutefois à l'objectif d'élimination et non de sélection.

Le cinquième chapitre met en place l'ensemble de la méthodologie pour allier synthèse de mécanismes et de consignes. Une solution optimale est extraite, basée sur la réalisation complète d'un suivi de trajectoire par l'organe terminal du manipulateur mobile. La finalisation du générateur de consignes est effectuée en déterminant un module de génération de configurations initiales générique, c'est-à-dire utilisant les performances face à la réalisation de la tâche et indépendant de la cinématique du robot commandé. Une synthèse des adaptations apportées au cours des différents niveaux de la problématique générale, ainsi que des améliorations qui en découlent, y est effectuée. Ces différentes améliorations tendent à augmenter d'une part la précision des solutions extraites des différentes optimisations (tant du point de vue de la synthèse de structures que de celle des consignes), d'autre part les performances temporelles des recherches effectuées et enfin permettent d'assurer l'émergence systématique d'une solution répondant au problème d'optimisation posé.

Chapitre 1

Conception robotique optimisée

La conception robotique consiste à déterminer les caractéristiques (variables de constitution) d'un système de manière à ce que celui-ci réponde aux exigences d'un objectif. Le système robotique est constitué d'une structure mécanique, d'actionneurs et d'un système de contrôle. La formalisation du problème de conception a permis d'extraire des méthodologies de conception basées sur plusieurs étapes. La conception préliminaire est l'une d'elles. Elle revêt un caractère essentiel en permettant de proposer des solutions optimales du point de vue d'un ou plusieurs critères pour un problème donné. Ce chapitre est destiné à définir les différents niveaux de notre problématique. Dans un premier temps le problème général de la conception est posé, puis plus particulièrement de la conception robotique. Les différents aspects rencontrés lors de la formulation du problème d'optimisation sont détaillés ainsi que quelques techniques d'optimisation numériques. Le problème particulier de la prise en compte de la commande d'un robot dans l'évaluation de sa structure lors de la phase de conception préliminaire est ensuite abordé. Enfin, l'approche que nous adoptons est exposée.

1.1 Conception optimisée

1.1.1 Formalisation du problème de conception

La conception optimisée correspond à la recherche d'un ensemble de variables répondant de manière optimale à un problème donné. Deux modèles essentiels peuvent être distingués pour la formalisation du processus de conception [Finger89] : l'approche descriptive et l'approche prescriptive.

Le modèle descriptif considère la conception comme un processus cognitif mis en œuvre par les humains et tente de le décrire. Il existe deux approches : la première consiste à formaliser le protocole de conception utilisé par les concepteurs [Maher87]. La deuxième tente d'émuler directement les habilités cognitives des concepteurs mis en

œuvre dans leur activité créatrice [Gero85]. Ce modèle est relativement subjectif et ne permet pas de dégager des processus cognitifs universels.

Le modèle prescriptif est un modèle comportemental. Il propose un modèle idéal soit du processus de conception [Juster85], soit de l'objet qui doit être conçu [Tomiyama89]. Ce comportement idéal fait l'objet de nombreuses divergences. Il peut être chaotique, organisé ou neutre. Cette méthode est difficilement évaluable car les performances d'un modèle ou d'un autre sont rarement étudiées de manière globale.

Ces deux approches, unies au progrès de l'outil informatique, ont permis l'émergence d'une méthodologie de la conception assistée par ordinateur. Elle consiste en l'élaboration de modèles informatiques particuliers, se concentrant plutôt sur la manière dont un ordinateur peut concevoir ou aider à la conception. Deux procédés se distinguent au sein de la CAO (Conception Assistée par Ordinateur) : ceux qui conçoivent et ceux qui analysent. Les premiers sont habilités à prendre des décisions tandis que les seconds extraient les informations utiles pour l'aide à la prise de décision. La CAO n'est pas réellement une nouvelle méthodologie pour la conception. Elle consiste plutôt en l'intégration d'outils d'analyse, d'évaluation et de synthèse sur une informatique. Trois méthodes de CAO sont dissociables :

- Par l'étude de cas : le stockage de solutions préconçues permet des propositions immédiates aux problèmes connus.
- A base de connaissances : utilise en plus de la précédente des heuristiques de conception appliquées au problème.
- Par l'optimisation : les décisions sont prises de manière itérative et conduisent à des solutions à partir de spécifications précises.

L'étude de cas est particulièrement efficace quand la décomposition du problème de conception est très poussée et que de nombreuses solutions sont disponibles. L'utilisation de bases de connaissances permet de l'améliorer mais ces deux premières méthodes restent limitées pour apporter des réponses à des problèmes nouveaux. L'optimisation reste la méthode la plus étudiée du fait des nombreuses possibilités potentielles qu'elle offre [Chirehdast94][Schmidt95][Kazuhiro95]. Elle présente cependant de nombreuses difficultés dont nous discuterons par la suite. Dans un premier temps Les étapes de conception sont décrites ainsi que les méthodes existantes pour l'optimisation.

1.1.2 Les étapes de conception

On distingue six étapes essentielles dans l'évolution du projet de conception. Elles ont été clairement identifiées lors de la mise en place de théories rigoureuses sur la manière dont un problème de conception doit être mené pour répondre à un besoin spécifique [Yoshikawa80][Ouezdou99]. Elles sont représentées sur la figure 4.2 et peuvent être détaillées de la manière suivante :

1. La spécification du besoin de la fonction à réaliser. Cette première phase consiste en l'établissement d'un cahier des charges précis.

2. La décomposition du problème en fonctions élémentaires va permettre une résolution "par morceaux" du problème global de conception.
3. La conception préliminaire, en utilisant des modèles réduits, est basée sur l'optimisation.
4. Le prototypage virtuel, ou la simulation réaliste du comportement de la solution courante, permet de visualiser le comportement de la meilleure solution issue de la conception préliminaire dans une simulation intégrant les contraintes de réalisation.
5. La construction d'un prototype de la solution retenue donne accès à une évaluation en situation réelle.
6. L'adaptation finale aux conditions réelles de fonctionnement permet, en utilisant les résultats de l'étape précédente, d'améliorer éventuellement la solution retenue.

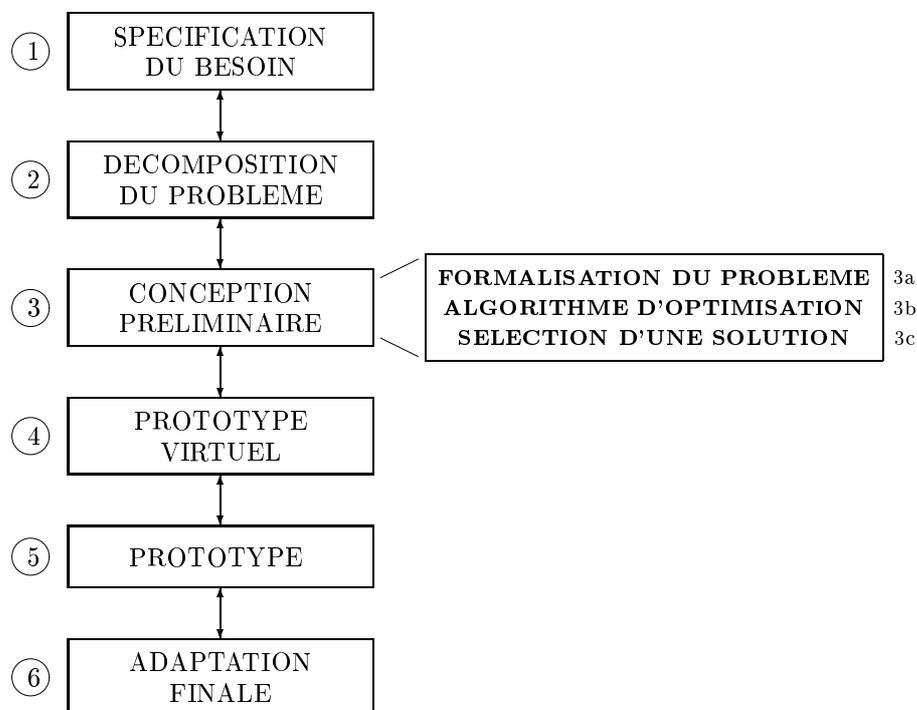


FIG. 1.1: Les différentes étapes de la conception.

Il peut être nécessaire de ré-examiner chaque phase lors de l'échec de l'une d'elles, ce qui donne une structure itérative au processus de conception. La CAO inclut les étapes 3 et 4. La conception préliminaire est l'étape qui nous concerne plus particulièrement, et elle peut être décomposée en trois sous-étapes qui seront décrites par la suite. Celles-ci sont (figure 4.2) :

1. la formalisation du problème sous la forme d'un problème d'optimisation,
2. l'utilisation d'un algorithme d'optimisation adapté,
3. la sélection d'une solution selon certains critères.

1.1.3 Optimisation pour la conception

L'optimisation pour la conception consiste à déterminer le meilleur système pour répondre à un problème donné. La manière dont le problème est posé est donc fondamentale et a provoqué l'émergence de plusieurs méthodologies. Ces méthodes basées sur l'optimisation progressent en synergie avec l'évolution de l'outil informatique. On distingue trois méthodes de conception par l'optimisation : paramétrique, configurationnelle et conceptuelle.

Dans la *méthode paramétrique*, la structure et les attributs (variables) de l'objet sont connus au départ et seules les valeurs de ces variables sont à déterminer. Elles peuvent prendre des valeurs numériques (réelles ou entières) ou autres (type d'un attribut). L'optimisation est généralement bien adaptée à cette méthode puisque le problème est totalement défini dès le départ. Malheureusement, la complexité de résolution de celui-ci ne permet pas souvent de les adapter.

La *méthode configurationnelle* transforme un concept physique en une configuration avec des attributs. Pour cette méthode, le problème principal réside dans la représentation et l'analyse de l'objet conçu [Libardi88]. En effet, la structure n'étant pas fixée, un travail important est à faire pour rendre le processus générique. La prise en compte des contraintes constitue également une difficulté majeure car elles interagissent de manière très circonstancielle selon la configuration [Cholvy83].

Enfin, pour la *méthode conceptuelle*, des fonctions désirées sont fixées. La recherche s'effectue sur toutes les structures pouvant réaliser ces fonctions d'une manière non spécifiée [Joskowicz88]. L'intérêt de cette méthode est qu'elle est plus générale que la précédente. Elle nécessite cependant de poser le problème d'une manière conceptuelle [Lee96].

Le choix entre ces trois approches ne dépend que de la nature et de la complexité du problème de conception. La conception robotique relève des mêmes règles. Pour bien distinguer les différents modèles, un exemple appliqué à ce domaine est présenté par la suite.

1.1.4 Conception robotique

Un système robotique est constitué par une structure mécanique, des actionneurs et un système de contrôle. Ces éléments sont généralement appelés variables de conception du robot [Chedmail98]. Ces sous-systèmes ainsi que l'environnement sont autant d'entités distinctes ayant leurs propres règles et comportements, mais interagissent en synergie. La complexité générée par leur interaction constitue un obstacle important à une modélisation simple et découplée des systèmes robotiques. En particulier, le couplage entre le système de commande et la structure commandée est tel qu'il doit être

pris en compte dès la phase préliminaire de la conception. C'est pourquoi il est important de considérer tous les aspects du robot (et de la tâche) et leur éventuelle évolution. Cependant, la conception robotique revient pour une grande part à faire de la synthèse de mécanismes [Chedmail98]. De nombreux modèles de conception mécanique pour la robotique ont été implémentés sur des systèmes informatiques [Regnier96][Chedmail98]. L'interaction structure/commande est une contrainte encore difficile à inclure malgré les progrès récents réalisés sur le plan de la méthodologie comme sur celui des outils de calculs mis à disposition. Du point de vue de la commande seule, un obstacle supplémentaire intervient. Il n'existe pas actuellement de commande à proprement parler qui puisse s'adapter sur des structures variables. Ce problème a pour conséquence l'impossibilité d'inclure la commande dans l'évaluation d'une structure dès la phase de conception préliminaire.

L'analyse des solutions et leur évaluation est donc réalisée indépendamment du comportement du robot. La conception robotique orientée tâche passe par une étape fondamentale d'optimisation des solutions. La difficulté majeure réside dans l'aspect automatique et objectif que doit revêtir le traitement des solutions, notamment lors de leur évaluation. La section suivante décrit le problème d'optimisation globale d'une manière générale.

1.2 Optimisation pour les systèmes robotiques

Un grand nombre de problèmes liés à la conception ou à la commande de systèmes complexes peuvent être formulés comme un problème d'optimisation globale [Adeli94]. Ces problèmes restent très difficiles à résoudre, autant sur le plan théorique que pratique, du fait de leurs spécificités et leur complexité. Nous présentons successivement le problème général de l'optimisation, les difficultés précédemment évoquées, les choix stratégiques à réaliser et les techniques de résolution associées. L'ensemble de ces techniques est passé en revue dans cette section. Une compréhension globale du problème d'optimisation est fondamentale pour notre objectif de conception et commande optimisées de robots.

1.2.1 Problème de l'optimisation globale

Nous allons dans un premier temps énoncer le problème d'optimisation globale sous contraintes afin d'en fixer les notations et le formalisme. La définition suivante est valable quel que soit le problème d'optimisation et nous nous y référerons tout au long de ce chapitre.

Définition 1 *Problème d'optimisation globale sous contraintes*

Soient \mathbf{E} l'espace de recherche avec $\dim(\mathbf{E}) = n$,
 $g_i : \mathbf{E} \Rightarrow \mathbf{R}$, $i \in \{1, \dots, q\}$ l'ensemble des q fonctions de contraintes,
 $\tilde{\mathbf{E}}$ un sous-espace de \mathbf{E} , tel que g_i satisfait : $\forall i \in \{1, \dots, q\}$, $(g_i(X \in \tilde{\mathbf{E}}) \geq 0)$
 $F : \mathbf{E} \Rightarrow \mathbf{R}$ la fonction coût d'optimisation.

Minimisation globale

Minimiser globalement $F(X)$, avec $X \in \tilde{\mathbf{E}}$, revient à :
 Trouver $X^* \in \tilde{\mathbf{E}}$, tel que $(\forall X \in \tilde{\mathbf{E}})$, $F(X^*) \leq F(X)$.
 X^* est appelé point minimum global et $F^* = F(X^*)$ minimum global.

Minimum local

Minimiser localement $F(X)$ avec $X \in \tilde{\mathbf{E}}$, revient à :
 Trouver $X^* \in \tilde{\mathbf{E}}$ tel que $(\exists \varepsilon > 0)$ $(\forall X \in B(X^*, \varepsilon) \cap \tilde{\mathbf{E}})$, $F(X^*) \leq F(X)$.
 X^* est appelé point minimum local et $F^* = F(X^*)$ minimum local.

Ici, le problème d'optimisation est défini comme une minimisation, mais la même définition peut être écrite dans le cas de la maximisation sans perte de généralité ou d'exactitude en utilisant la relation :

$$\max \{F(X) | X \in \tilde{\mathbf{E}}\} = - \min \{F(X) | X \in \tilde{\mathbf{E}}\}$$

Aucune spécification de l'espace de recherche \mathbf{E} n'est nécessaire, ce qui signifie que l'optimisation peut être menée sur un espace quelconque (continu, discret ou booléen). La seule contrainte est que la fonction objective F puisse trouver une valeur et une seule pour tout point $X \in \mathbf{E}$, ce qui signifie que F doit être une application de \mathbf{E} dans \mathbf{R} .

En ce qui concerne les équations de contraintes g_i , elles peuvent être généralisées par des processus déterministes quelconques comme des équations mathématiques (implicites, numériques ou booléennes) ou des algorithmes (par exemple d'optimisation déterministe). Si on considère le cas d'une optimisation effectuée dans le cadre de la détermination d'une configuration optimale pour un manipulateur mobile, des contraintes "molles" sont plus généralement utilisées : elles peuvent être violées moyennant une pénalisation introduite dans la fonction objective F à optimiser.

Dans les problèmes d'optimisation paramétriques, l'espace de recherche \mathbf{E} est un sous-espace de l'espace Euclidien avec $\mathbf{E} \subseteq \mathbf{R}^n$, ce qui simplifie considérablement le problème. Dans ce cas, les techniques d'optimisation adaptées sont décrites dans plusieurs ouvrages [Minoux83]. Il a été montré que sur l'ensemble des problèmes d'optimisation, toutes les méthodes se valent [Wolpert95]. Malgré cela, il apparaît que certaines sont plus appropriées que d'autres en fonction du cas étudié [Macready95]. Il faut donc choisir une méthode adaptée au problème traité.

1.2.2 Spécificités de l'optimisation

L'optimisation est largement utilisée de nos jours dans la conception ou la commande de systèmes complexes. Grâce à l'émergence de moyens de calculs puissants, des problèmes d'optimisation insolubles il y a 20 ans peuvent aujourd'hui être abordés. Certains restent encore difficiles dans leur résolution du fait même de la problématique d'optimisation, mais aussi de la complexité des systèmes traités. Dans un premier temps, pour que le problème d'optimisation soit bien posé, il faut déterminer clairement trois points :

- La définition précise des *variables à optimiser*, celles *fixées*.
- L'établissement des *contraintes* à considérer.
- La *stratégie d'optimisation*.

Constitution des solutions

La première étape dans un problème d'optimisation est de déterminer l'ensemble des paramètres qui constituent l'objet de la recherche. Contrairement à l'optimisation paramétrique où cet ensemble est posé dans la formulation même du problème, notre approche implique des choix :

a) Domaine d'optimisation : Les parties qui vont être optimisées et celles qui sont fixées doivent être déterminées. Il est théoriquement possible de choisir une paramétrisation globale et complète du système, mais cela suppose une généralité absolue des solutions réalisables, ce qui est pratiquement irréaliste. Il est possible à l'inverse de se restreindre à une structure particulière préconçue (système de locomotion fixé par exemple) et effectuer une recherche partielle des paramètres de conception (le nombre des organes de locomotion ou leur point d'attache sur le corps du robot). Il semble raisonnable d'opter pour une stratégie intermédiaire en choisissant des sous-structures existantes paramétrables et en leur permettant de se combiner librement lors de l'optimisation. Il est néanmoins indispensable de choisir ces paramètres en fonction de leur pertinence par rapport aux fonctions et performances recherchées.

b) Paramètres non valués : Les paramètres qui découlent des précédents choix ne sont pas nécessairement faciles à manipuler par un programme d'ordinateur qui ne considère finalement que des entités mathématiques. Un exemple typique en conception robotique est le type cinématique d'une liaison (rotoïde ou prismatique). Interpréter ce type de paramètres n'est pas immédiat au sens mathématique et demande un traitement spécifique (modification du modèle géométrique). Sans ce traitement, il est difficile de qualifier l'impact de ces paramètres sur le fonctionnement global du système robotique. Cela constitue l'un des problèmes majeurs de l'optimisation des systèmes complexes en général et des systèmes mécatroniques en particulier.

c) Couplage et redondance : Il est assez fréquent d'avoir des couplages importants entre différents paramètres d'une même structure. Dans le cas des robots

manipulateurs, il suffit de modifier la valeur angulaire d'une articulation pour avoir un effet significatif sur la plupart des critères d'évaluation. Cela milite pour l'intégration d'absolument tous les paramètres susceptibles d'influer sur le résultat global dans l'optimisation. Ces couplages peuvent également cacher des redondances et multiplier inutilement le nombre de ces paramètres, gênant et compliquant la recherche de solutions optimales. Il suffit par exemple, en cinématique, de composer successivement deux rotations de même axe pour avoir une redondance des deux angles de rotation du point de vue de la rotation globale. Prendre en compte les couplages et éviter les redondances de représentation constitue un travail délicat à réaliser pour chaque problème traité.

Fonction d'évaluation

Tout processus d'optimisation fait appel à une fonction d'évaluation du ou des critères d'optimisation que l'on appelle *fonction objective*. Cette fonction peut être très simple, comme la valeur d'une fonction à optimiser, ou très complexe, comme la performance d'un robot marcheur. La détermination de cette fonction relève uniquement du domaine d'étude et elle est totalement tributaire du modèle de quantification de la performance choisie. Cette fonction objective doit donc avoir certaines qualités intrinsèques rendues indispensables par l'optimisation. Ces qualités sont la pertinence, la robustesse et la rapidité d'évaluation. Nous citons ci-après les difficultés liées à cette fonction qui rendent son optimisation problématique.

a) Irrégularités de la fonction objective : Un problème fréquemment rencontré dans l'utilisation de cette fonction concerne ses irrégularités. Les méthodes d'optimisation classiques utilisent généralement des fonctions auxiliaires liées à la fonction objective pour orienter leur recherche, comme par exemple ses dérivées successives. Bien souvent, dans le cas des systèmes robotiques, les fonctions utilisées sont fortement irrégulières. A titre d'exemple, les modèles géométriques, cinématiques, dynamiques sont rendus fortement irréguliers par les contraintes technologiques (butées articulaires), les limites d'accessibilité (singularités) ou les différents types d'articulations. Ces irrégularités peuvent s'exprimer sous la forme de non-linéarités, discontinuités, zones interdites, hystérésis ou pics isolés.

b) Problématique multi-critères : Pour évaluer les performances du robot, on utilise généralement plusieurs critères. Ceux-ci peuvent être mathématiques ou autres, résulter de la simulation par exemple. Dans un problème d'optimisation, le formalisme (définition 1) force la définition d'une fonction objective à valeur unique. Rassembler tous les critères en une seule fonction objective soulève la question du choix multi-critères. Une moyenne de ces critères est mathématiquement accessible, ou alors le parti d'une optimisation réellement multi-critères peut être pris (optimisation de *Pareto* [Obayashi97]). Cette dernière option demande de prendre des décisions dans la boucle d'optimisation.

c) Défaillances de la fonction objective : Lors de la construction d'une fonction objective, il est essentiel de soulever le problème de sa pertinence par rapport à son objectif. Dans certains cas, la fonction remplira mal son rôle de représentation d'objectif et l'évaluation des solutions sera erronée. Ce problème est d'autant plus susceptible d'apparaître lors de l'utilisation d'outils d'évaluation complexes tels que la simulation avec les problèmes de divergence numérique par exemple.

Stratégies de recherche

Quelle que soit la méthode d'optimisation utilisée, plusieurs choix stratégiques liés à la nature du problème d'optimisation robotique et à sa mise en œuvre sont disponibles. Ceux-ci sont essentiellement liés à la taille de l'espace de recherche et le souci de déterminer un optimum global au problème posé.

a) Exploration et exploitation : Le dilemme qui se pose pour toute optimisation est de doser l'effort de recherche (exploration) et la tentation de consommer (exploitation). Entre deux solutions, la plus performante peut immédiatement séduire l'utilisateur, ce qui peut mener à un minimum local. Ce phénomène de convergence prématurée est considéré indésirable puisque la recherche est axée sur la détermination d'un minimum global. Le problème inverse consiste à changer constamment d'objectif et à réaliser une errance permanente dans le domaine de recherche sans jamais converger. Un juste équilibre entre ces deux caractéristiques est à déterminer et éventuellement à ajuster au cours de l'optimisation.

b) Optimisation globale : Considérant l'ensemble des paramètres à optimiser pour un système robotique (variables structurelles et comportementales), il existe deux grandes méthodologies d'optimisation. La première est appelée séquentielle et consiste à scinder le problème global en plusieurs sous-problèmes (modularité) et à les résoudre indépendamment. Un sous-problème peut utiliser les résultats des précédents, ou il peut en être indépendant [Kim92][Paredis96]. Cette méthode comporte deux inconvénients majeurs liés à la division du problème global en sous-problèmes semi-indépendants. Le premier est l'ordre de résolution des sous-problèmes. Chacun d'eux ne doit dépendre que des solutions aux sous-problèmes situés en amont dans la séquence de résolution. Le second apparaît si l'on considère le cas où les sous-problèmes sont fortement couplés. De sérieuses concessions sont alors à envisager sur la globalité de l'optimisation. Bien sûr, l'optimisation séquentielle est tout à fait valable dans le cas où le sous-problème primaire consiste à supprimer les solutions inadaptées (satisfaction des contraintes). Cela permet de restreindre la recherche dans un sous-domaine contenant uniquement les solutions éventuellement exploitables.

c) Interdépendance de la structure et de la commande : Mise à part le respect des contraintes technologiques, il n'existe pas vraiment de critères strictement structureaux ou comportementaux dans un problème robotique. Il est donc extrêmement difficile d'évaluer la structure mécanique et la commande d'un robot indépendamment.

En dehors de l'étude expérimentale, la seule évaluation objective possible peut se faire à travers la simulation du système robotique complet. L'évaluation d'un système complexe doit donc utiliser une fonction objective qui inclue ces deux sous-parties.

1.2.3 Méthodes d'optimisation

Les méthodes d'optimisation globales ou locales, avec ou sans contraintes, peuvent se classer en deux catégories : déterministes ou stochastiques. Les méthodes déterministes consistent à partir d'une solution et à appliquer des heuristiques d'optimisation pour trouver une nouvelle solution. On procède d'une manière itérative jusqu'à la convergence de la méthode. Les méthodes stochastiques génèrent des nombres aléatoires pour trouver de nouvelles solutions. Ces dernières peuvent être associées à des heuristiques de recherche déterministes pour donner des méthodes hybrides [Gascuel96].

Philosophie des méthodologies

Les *méthodes déterministes* utilisent des règles de recherche générales et invariables, ce qui les conduit à se comporter d'une manière déterminée. Une solution consiste à utiliser des méthodes dites exactes qui vont directement calculer le minimum (ou maximum) global selon un principe général (par exemple l'annulation de la dérivée). Malheureusement, de telles méthodes deviennent très vite inapplicables (besoin de fonctions analytiques) et ne sont disponibles que pour une classe très réduite de problèmes.

Une autre approche consiste à appliquer itérativement des heuristiques particulières ou générales, c'est-à-dire des connaissances particulières au problème traité ou générales à l'optimisation. L'exemple le plus connu est la méthode de descente de gradient généralisée qui dirige la recherche localement selon la meilleure descente disponible [Griewank81]. Cette méthode nécessite, d'une part, l'existence et la calculabilité d'un gradient de la fonction objective, et d'autre part, un point de départ pour initialiser la recherche. Le problème est que le minimum trouvé est probablement local et qu'il dépend du point de départ. Les méthodes déterministes ne sont donc initialement ni générales, ni globales. Cela signifie que leur utilisation n'est pas toujours possible et que les solutions trouvées ne sont que localement optimales. Pour la plupart des problèmes d'ingénierie, une bonne solution est satisfaisante même si elle n'est pas optimale. C'est pourquoi les méthodes déterministes heuristiques sont largement utilisées actuellement pour résoudre les problèmes d'optimisation globale.

Les *méthodes stochastiques* font intervenir des processus aléatoires dans la recherche des solutions optimales. Ces méthodes sont itératives et sont souvent constituées en deux phases. La première est globale et consiste à échantillonner aléatoirement l'espace de recherche afin de créer des solutions initiales. La deuxième phase utilise les solutions initiales et recherche dans leur voisinage de meilleures solutions. La procédure de recherche peut être aveugle et mener à une recherche aléatoire pure [Rinnooy85], ou elle peut être dirigée selon des heuristiques d'optimisation [Price87]. Ces méthodes

peuvent être rendues très efficaces par l’adjonction d’heuristiques performantes mais la difficulté est de garantir la fiabilité en ce qui concerne la globalité de la recherche (la complétude).

Méthodes déterministes

Les méthodes déterministes sont les plus anciennes et utilisent généralement les premiers résultats du calcul différentiel. Elles ont au cours des siècles montré leur puissance et leurs limites. Nous pouvons citer quelques exemples classiques de méthodes :

- le Gradient généralisé :

$$X_{i+1} = X_i + \delta X \times \nabla F(X_i)$$

X_{i+1} est l’extrapolé linéaire de X_i , δX le pas d’itération et $\nabla F(X_i)$ le gradient de la fonction d’évaluation F en X_i . Cette méthode n’est applicable que sur un espace discret.

- le Hill-Climbing, utilisant aussi les principes du gradient :

$$X_{i+1} = \max(X \in B(X_i, \varepsilon))$$

X_{i+1} est le meilleur voisin de X_i . Cette méthode s’applique dans tout espace naturel sans condition sur la fonction F . Elle reste une méthode locale (détermine le plus proche optimum local) et coûteuse.

- Enumérative ou Exhaustive : tout l’espace de recherche est parcouru. La recherche aboutit donc sur un optimum global, mais n’est utilisable que dans le cas d’espaces finis et dénombrables, avec un coût maximal.

Méthodes	Espace	Fonction	Portée	Coût	Connaissances
Gradient	continu	régulière	locale	faible	X_0 bien choisi
Hill-Climbing	quelconque	quelconque	locale	fort	X_0 bien choisi
Enumérative	fini et dénombrable	quelconque	globale	maximum	grille bien choisie

TAB. 1.1: Caractéristiques des méthodes déterministes

Ces méthodes sont longuement développées dans certains ouvrages de références, par exemple [Minoux83]. Leurs caractéristiques générales sont exposées dans le tableau 1.1, où leur caractère local apparaît, ou au moins leur limite dans le type d’espace de recherche. De plus, elles utilisent -pour les plus efficaces d’entre elles- des connaissances a priori de la fonction objective F . Il existe de nombreuses variantes de ces méthodes, comme la méthode de Newton-Raphson. Certaines s’affranchissent d’une partie des inconvénients cités mais rarement de tous. Certaines techniques de dépôts multiples permettent de rendre la recherche plus globale. Elles sont assez inefficaces en termes de temps de calcul car les différents minima locaux sont retrouvés de nombreuses fois. Les techniques sophistiquées de regroupement permettent d’éviter ce problème, mais elles restent largement dépendantes de la dimensionalité de l’espace de recherche [Dixon76].

Méthodes Stochastiques

Les méthodes stochastiques sont plus récentes et font appel à des opérateurs du même nom qui introduisent une composante aléatoire dans la recherche [Dufflo96]. Le développement de l'outil informatique depuis cinquante ans a permis d'utiliser ces méthodes à grande échelle et de les appliquer à des problèmes très complexes [Rinnooy87]. Ces méthodes peuvent être hiérarchisées en deux groupes :

- **Méthodes aléatoires**
 - Monte-Carlo
 - Recherche Taboue
- **Méthodes naturelles**
 - Recuit Simulé
 - Evolutionnaires

Les *méthodes aléatoires* font appel dans leur recherche uniquement à une heuristique forte : *quelle que soit la meilleure solution trouvée, la meilleure solution globale peut être n'importe où ailleurs*. Ainsi, la méthode de Monte-Carlo consiste à tirer la nouvelle solution au hasard avec une loi uniforme sur l'espace de recherche. La Recherche Taboue consiste, en plus d'un tirage aléatoire, à éviter de retomber sur une solution déjà envisagée en mémorisant le chemin déjà parcouru [Glover77].

Les *méthodes naturelles* s'appuient sur des phénomènes naturels qui présentent des aspects intéressants pour l'optimisation. Le Recuit Simulé modélise le phénomène de refroidissement des métaux à la suite de traitements thermiques [Kirkpatrick83] [Schmidt95]. Les atomes de métal sont fortement excités lors de l'échauffement, puis leur niveau d'excitation diminue graduellement lors du refroidissement en prenant des états intermédiaires plus ou moins stables. La loi de changement d'état est une loi Boltzienne qui diminue la probabilité de changement d'état au fur et à mesure que la température décroît [Hajek88]. Ainsi, la méthode converge vers un puit de potentiel lorsque la température n'est plus suffisante pour franchir le mur que constitue le gain d'énergie demandé pour en sortir. L'optimisation que la nature nous offre ici est celle de la minimisation de l'énergie interne du métal. Cette minimisation n'est malheureusement pas complètement globale puisqu'un état de minimum local peut être atteint et conservé. Le point de départ de la recherche est primordial car il détermine en grande partie le puit de potentiel vers lequel la méthode va converger. La configuration initiale, la température initiale, sa fonction de décroissance et le critère d'arrêt de la recherche sont choisis en fonction du problème. Rapidité de convergence et qualité de la solution en découlent.

Les méthodes Evolutionnaires utilisent des algorithmes qui simulent l'évolution naturelle (génétique) des populations d'êtres vivants. La méthode associe la théorie de l'évolution darwinienne (survie du plus adapté, [Darwin59]) aux lois de la génétique (reproduction, croisement, mutation, [Mendel65]) pour mettre en place et exploiter

un modèle d'évolution artificielle [Holland75]. L'application des algorithmes évolutionnaires à l'optimisation globale a connu un essor important ces dernières années en raison de leurs qualités intrinsèques [Goldberg89][Michalewicz92][Back96]. Ils permettent de faire évoluer une population de solutions vers le but recherché. Le parallèle entre le processus naturel et l'optimisation de systèmes robotiques autonomes est facile à établir puisque l'objectif, les environnements et les contraintes sont similaires. En effet, le but final est l'adaptation de systèmes naturels (faune, flore) à des environnements dynamiques (changements de saisons, catastrophes naturelles) sous contraintes technologiques (chimiques, physiques) et de ressources (énergie, matière). Les méthodes

Méthodes	Espace	Fonction	Portée	Coût
Monte-Carlo	quelconque	quelconque	globale	extrême
Recherche Taboue	quelconque	quelconque	globale	extrême
Recuit Simulé	quelconque	quelconque	globale	fort
Evolutionnaires	quelconque	quelconque	globale	moyen

TAB. 1.2: Caractéristiques des méthodes stochastiques

naturelles sont donc en principe globales (avec certaines réserves pour le Recuit Simulé) et peuvent aborder tout problème d'optimisation, quelle que soit la nature de la fonction objective F (tableau 1.2). Elles permettent donc une optimisation ambitieuse là où les méthodes déterministes sont souvent impuissantes ou très limitées. Les méthodes purement aléatoires sont certes plutôt brutales et primaires, mais permettent dans des cas extrêmes de faire la meilleure optimisation possible là où les autres ne font que perdre plus de temps. Les Algorithmes Evolutionnaires combinent des opérateurs stochastiques et déterministes, ce qui permet de réaliser une optimisation à la fois globale et minimale. L'inconvénient majeur de la méthode du recuit simulé est le temps nécessaire pour l'aboutissement de l'optimisation (en pratique, la vitesse de décroissance de la température est accélérée, ce qui augmente les chances de conclure sur un optimum local).

1.3 Le problème de la commande dans la conception préliminaire robotique

1.3.1 Problème général

La commande d'un système robotique consiste à adapter le comportement du système de telle sorte qu'il converge, quel que soit son état, vers un état désiré. La performance d'une loi de commande est caractérisée par sa précision et sa robustesse.

- La *précision* assure que le résultat obtenu correspond bien à ce que l'on désire.
- La *robustesse* assure ce résultat même si le système subit des perturbations.

Un équilibre est donc à déterminer entre ces deux caractéristiques qui varient inversement proportionnellement l'une à l'autre. Une très grande précision sera obtenue au

dépend de la robustesse et réciproquement : une loi de commande hautement adaptable au système commandé sera peu regardante sur la précision.

Il existe de nombreuses lois de commande dont la simplicité de mise en œuvre et l'efficacité dépendent de la complexité du système commandé [Saadia00]. Cependant, quasiment toutes exigent de grandes connaissances de ce système pour être performantes. De nombreuses recherches dans le domaine ont permis la mise en place de commandes qui peuvent s'adapter aux variations du système ou de l'environnement : commandes robustes, passives, adaptatives [Saadia99]... Elles sont utilisables sous de fortes contraintes, la plus grande étant liée au caractère linéaire (ou linéarisable) du système commandé. Aucune cependant ne peut être définie pour un système a priori inconnu, ou devant subir de très grandes modifications, tout en maintenant une précision "acceptable". Or c'est le cas dans lequel se trouvent les robots lors de la phase de conception préliminaire, puisque leur structure (type et nombre des liaisons par exemple) évolue. C'est pourquoi le comportement général du robot sur toute la réalisation de la tâche n'est pas pris en compte lors de cette phase. Le prototypage virtuel d'une solution tente de remédier à ce problème en adaptant une commande particulière à un robot localement adapté et de dynamique connue avant la fabrication d'un prototype réel. Si cette étape n'est pas concluante, la précédente est renouvelée.

1.3.2 Méthodologies existantes

Les modèles de commande ont montré leurs limites dans un problème de conception préliminaire. La mise en place d'une commande "générique", c'est-à-dire pouvant s'appliquer quel que soit le système considéré, a fait l'objet d'études sporadiques. Ces commandes sont envisagées lors de la conception préliminaire. Les méthodologies mises en place à ce jour sont encore titubantes et aléatoires. Elles font toutes appel aux méthodes d'intelligence artificielle ou évolutionnaires. Les algorithmes évolutionnaires ont été utilisés dans le cas de la conception préliminaire de plate-formes mobiles [Chocron00] pour réaliser une tâche simple, comme aller le plus loin possible de la position de départ ou tourner le plus vite possible autour de l'axe vertical. La consigne est mise sous la forme

$$U_j = \alpha_j * \cos(\omega_j * t + \varphi_j)$$

- U_j : Tension appliquée à la liaison L_j
- α_j : Amplitude de U_j
- ω_j : Pulsation de U_j
- φ_j : Phase de U_j
- t : Temps de la simulation

Ce type de fonction sinusoïdale pour les consignes représente la consigne à appliquer à chaque actionneur modélisé par un moteur à courant continu avec son réducteur. L'évaluation des consignes est faite sur la totalité de la simulation, elle dépend uniquement de la performance du robot (distance parcourue ou vitesse de rotation). Cette

méthode présente l'avantage de ne nécessiter aucune information sur le système en mouvement. Cependant, son inconvénient majeur est de ne recevoir non plus d'information sur l'environnement ou la tâche à accomplir. Son fonctionnement est totalement aveugle et ne constitue donc pas à proprement parler une commande dans le sens où l'absence de retour n'autorise pas la réalisation d'une tâche (suivi de trajectoire par exemple). Cette génération de consignes relève donc plus du contrôle : le contrôle d'un système peut être vu comme un observateur. Contrairement à la commande, il n'intervient pas pour modifier le comportement du système en cas d'écart avec l'état désiré mais indique si l'objectif a été atteint. La méthode a cependant donné des résultats positifs, utilisée dans le cadre de l'optimisation structurelle de robots mobiles devant répondre à des tâches simples du type "maximiser la vitesse de rotation du robot" ou "maximiser son altitude".

L'utilisation de cette technique de génération de consignes ne peut susciter un intérêt que si l'erreur entre l'état désiré du système (situation, vitesse, effort... désirés) et l'état réel correspondant constitue un critère d'évaluation à minimiser. Ainsi une trajectoire pourra être suivie ou un effort appliqué. Ce type de commande soulève des problèmes, par exemple l'évolution des consignes avec l'avancement dans la réalisation de la tâche, les manœuvres éventuelles à effectuer. Un problème supplémentaire d'une telle approche est que la complexité du système à contrôler augmente si rapidement qu'aucune solution n'émerge. La détermination de la commande d'un robot en phase de conception préliminaire ne peut donc qu'être séparée de l'optimisation de la structure tout en s'adaptant à son évolution.

1.3.3 Le comportement dans l'évaluation de la structure

L'interaction entre la structure d'un robot et la commande de cette structure est évidente. Il est donc nécessaire d'inclure cette commande dans la boucle d'optimisation de la structure pour obtenir un robot adapté à la tâche qu'il doit effectuer. En général, pour être sûr de la performance de la structure d'un robot, des marges de sécurité très larges sont utilisées. Ceci a pour résultat des machines sur-performantes par rapport à la tâche, donc sous-exploitées du point de vue de leur capacités. Certains avantages résultent de cette situation : des possibilités d'adaptation à des tâches plus complexes si nécessaire.

Notre objectif est donc de mettre en place une méthodologie de conception robotique structurelle et comportementale. Il présente de grandes difficultés comme nous l'avons évoqué dans les sections précédentes. La méthodologie est appliquée à des structures adaptées et en pleine évolution : les manipulateurs mobiles. Elles sont adaptées à notre problème car difficilement évaluables localement, comme des manipulateurs seuls ou des plate-formes mobiles isolées. Le caractère plus élaboré d'une tâche de manipulation mobile *nécessite* une évaluation sur la réalisation globale de la tâche.

Nous présentons dans un premier temps la problématique de la manipulation mobile. L'approche que nous proposons est ensuite détaillée.

1.4 Manipulateurs mobiles

Cette partie précise la problématique générale de la manipulation mobile. Un état de la recherche plus complet dans ce domaine est présenté en annexe 1.

1.4.1 Robots manipulateurs mobiles

Un manipulateur mobile résulte de l'assemblage d'un bras manipulateur et d'une plate-forme mobile. Cette association présente donc de nombreuses possibilités puisque la complémentarité de deux robots différents à tout point de vue permet l'élargissement du champ d'action de chacun, tant au niveau de la définition ou du degré de complexité de la tâche que de sa réalisation. Les avantages de ce système touchent par exemple à l'évitement des singularités, le contournement d'obstacles, l'augmentation de l'espace de travail, la coopération.

Si ces améliorations sont particulièrement attrayantes, elles sont accompagnées d'une grande complexité de réalisation. De nombreuses études ont déjà été effectuées sur la commande des bras manipulateurs et des engins locomoteurs robotisés et des solutions aux problèmes respectifs ont été apportées. Cependant l'association met en œuvre un très grand nombre de paramètres dont l'étude de la gestion est toujours d'actualité. Quoi qu'il en soit, la détermination d'une commande fait systématiquement appel à l'optimisation de critères qui varie avec les impératifs liés à la tâche (Annexe 1).

En revanche, le problème de conception de tels robots n'a pas encore été étudié. En effet les deux tâches (de manipulation et de déplacement) sont considérées comme totalement découplées du point de vue de la conception. Il suffit donc, pour obtenir un manipulateur mobile efficace du point de vue de sa structure, d'étudier une base mobile adaptée au déplacement dans l'environnement désiré et un bras manipulateur adapté à la tâche de manipulation.

Problématique de la manipulation mobile

Les possibilités offertes par cette famille robotique ne font aucun doute. Ses avantages sont essentiellement liés à l'élargissement du champ d'action de chaque robot. La mise en pratique présente quelques difficultés encore d'actualité. Certaines sont évidentes puisque déjà rencontrées lors de l'étude des manipulateurs ou des plate-formes mobiles indépendamment. Par exemple, le respect des contraintes de non-holonomie pour la base mobile ou d'évitement des singularités cinématiques pour le bras manipulateur. A ces contraintes "classiques" s'ajoutent de nouveaux problèmes du fait de l'association des deux sous-robots :

- L’augmentation conséquente du degré de redondance du système
- Les interactions dynamiques
- La complémentarité des deux robots
- La stabilité du système

La redondance traduit le fait qu’à une position de l’organe terminal du manipulateur (l’effecteur) correspond une infinité de positions possibles pour le corps du robot. Mathématiquement le modèle du robot n’est pas inversible directement. Ce problème apparaissait déjà dans le cas de certains bras manipulateurs, il est à présent augmenté des degrés de liberté de la base mobile (trois de plus pour un déplacement plan par exemple).

Le problème lié à l’utilisation complémentaire des deux robots rejoint celui de la redondance, à savoir comment utiliser la base mobile pour que l’aide à la manipulation soit effective. Les interactions dynamiques n’interviennent bien sûr que dans le cas d’une étude en dynamique : les accélérations de chaque robot vont créer des forces agissant sur le robot partenaire, ces forces n’étant plus négligeables à partir d’un certain niveau d’accélération. La stabilité et la précision de déplacement de l’effecteur s’en ressentent.

Enfin, la stabilité dynamique du système est à considérer comme un problème à part entière lors de déplacements rapides ou de la manipulation d’un objet lourd, afin d’éviter le renversement du manipulateur mobile. Les différentes approches proposées dans la littérature pour l’étude et la résolution de ces problèmes sont décrites en annexe 1. Ces différents problèmes ont mené les recherches vers un découplage systématique en sous-problème de manipulation et de déplacement, plus simples à gérer. Cette décomposition est décrite ci-après.

Découplage classique du problème

Toutes les études sur le problème de la manipulation mobile considèrent le manipulateur et la plate-forme mobile comme deux entités distinctes. Les problèmes qui leurs sont liés sont gérés séparément à tout moment. Ceci semble logique du fait des bonnes connaissances acquises sur chacun de ces deux robots. La figure 1.2 résume l’approche généralement faite pour aborder le problème de manipulation mobile dans la recherche et met en évidence cette dissociation.

On peut voir cet aspect de manière beaucoup plus général en considérant qu’un problème complexe de robotique autonome est abordé via le concept de modularité. La modularité consiste à diviser un problème général en sous-problèmes plus simples à gérer dans une approche séquentielle. La mise en place d’interfaces pour que tous ces sous-problèmes soient pris en compte dans la solution finale est ensuite nécessaire. L’indépendance effective de chaque sous-problème n’est malheureusement pas *systématiquement acquise*, elle est parfois même complètement ignorée. Le fonctionnement du système général est souvent assuré par l’intervention humaine ou la mise en place

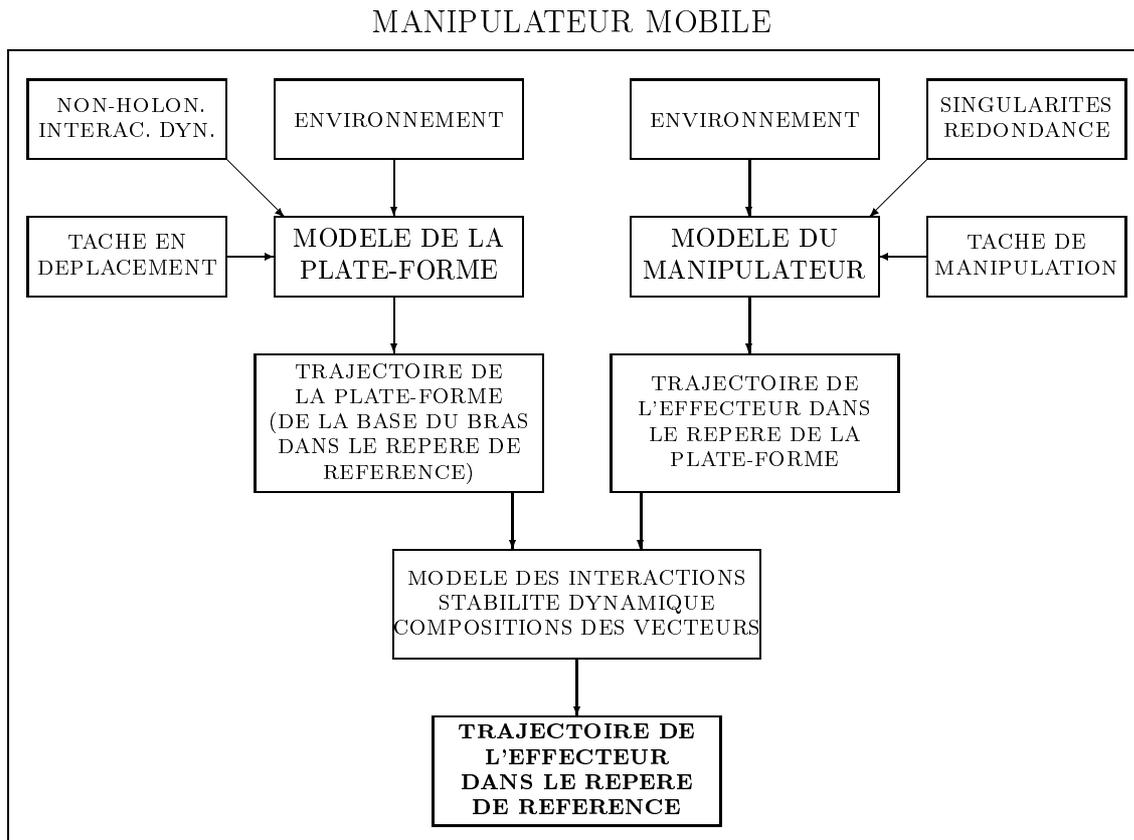


FIG. 1.2: Approche dissociée de la manipulation mobile.

de conditions strictes d'utilisation. Un exemple évident existe avec les manipulateurs mobiles : les interactions dynamiques. Très peu abordées dans la recherche, elles n'en restent pas moins un problème réel dès que l'étude sort du cadre "quasi-statique", c'est-à-dire dans la plupart des cas où l'intervention de tels robots sera sollicitée.

Remarquons toutefois que la modularité part d'un problème complexe et le subdivise pour le simplifier. Le problème de manipulation mobile présente historiquement le schéma inverse : deux problèmes plus simples sont réunis pour en former un complexe. De ceci ressort une double subdivision : la décomposition agit sur la conception (séparation de la structure et de la commande) et sur la tâche (considération séparée du déplacement et de la manipulation). Il y a donc quatre sous-problèmes à résoudre à la base même de l'étude d'un manipulateur mobile. Si ces décompositions successives sont justifiables de par la très grande redondance du système et par la nature hétérogène des paramètres et des contraintes qui lui sont liés, elles impliquent à chaque niveau une part importante de décision humaine, donc augmentent considérablement le degré de subjectivité dans la solution finale. De plus, comme évoqué précédemment, l'indépendance complète entre les sous-problèmes est loin d'être évidente et systématique. Il est donc nécessaire de reprendre le problème global et de refaire les séquences d'études. Un manipulateur ne peut plus être considéré comme une simple extension

de base mobile -et réciproquement une base mobile comme un outil de déplacement pour manipulateur. Le manipulateur mobile doit être vu dès le départ comme un robot unique devant résoudre une tâche globale de manipulation mobile.

1.5 Approche proposée

1.5.1 Conception préliminaire orientée tâche

Notre approche consiste à considérer le problème de manière globale. Certains choix sélectifs sont à effectuer avant de se lancer dans une telle entreprise, l'état actuel des connaissances ne permet pas de définir directement l'ensemble des paramètres dans un espace de recherche aussi vaste. La décomposition pour laquelle nous optons est orientée vers les étapes de réalisation de la tâche et non sur la tâche elle-même. Une technique pyramidale inversée de conception est adoptée, elle est illustrée sur la figure 1.3. Celle-ci permet d'éliminer au cours de la sélection les robots qui ne peuvent pas, cinématiquement, réaliser la tâche. Deux étapes peuvent alors être envisagées (Fig.

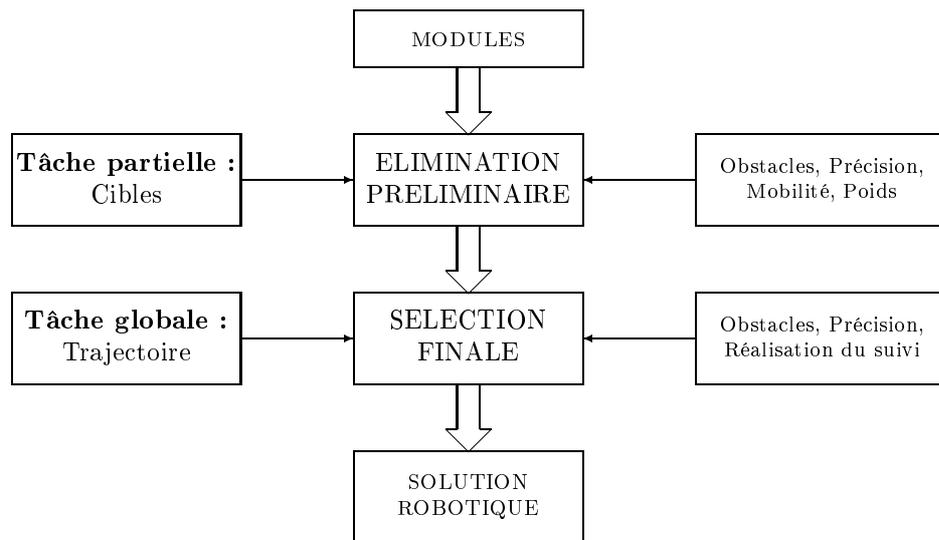


FIG. 1.3: Conception préliminaire orientée tâche d'un manipulateur mobile.

1.3), dont les détails vis-à-vis des variables à optimiser figurent dans le tableau 1.3 :

1. Une sélection préliminaire qui permet d'*éliminer* les topologies non adaptées cinématiquement. La tâche globale est décomposée de manière à tester les robots sur seulement des parties de celle-ci et l'étude est effectuée sans déplacements. Les structures adaptées sont enregistrées ainsi que les meilleures configurations correspondantes et la valeur des performances des robots. Cette étape a pour objectif de restreindre la recherche uniquement aux structures pouvant potentiellement réaliser la tâche. Ceci évite de tester la commande sur des systèmes non aptes et économise donc du temps de calcul.

2. L'étape finale qui étudie les robots restants. L'ensemble des paramètres de conception est pris en compte et la tâche globale est considérée pour une optimisation structurelle et configurationnelle. Cette étape consiste à *perfectionner* des topologies dont les compétences face à la tâche ont été vérifiées. Les robots ayant passé avec succès l'étape précédente sont testés sur la globalité de la tâche avec une commande qui prend en compte leurs caractéristiques dynamiques. Encore une fois, *élimination* des robots qui n'y parviennent pas.

	Elimination préliminaire	Génération de consignes	Sélection finale
Représentation cinématique	Denavit-Hartenberg	id	id
Paramètres articulaires	Type, axes	-	-
Paramètres Structurels	ℓ_i , nombre de ddl	-	-
Paramètres Configurationnels	x_b, y_b, θ $\phi_1, \dots, \phi_{n_b}$	$x_b(t), y_b(t), \theta(t)$ $\phi_1(t), \dots, \phi_{n_b}(t)$ + Config. init.	id
Critères de tâche	Positionnement local (ponctuel)	Positionnement global (trajectoire)	id
Critères de Performance	Précision, obstacles, struct. min., mobilité	Précision, obstacles réalisation du suivi	id + ddl min

TAB. 1.3: Détermination progressive des paramètres.

Cette approche permet de restreindre l'ensemble des solutions robotiques existantes aux seules solutions capables de réaliser la tâche, sans restreindre l'espace de recherche de ces solutions. Elle n'est envisageable que si les contraintes appliquées au système sont minimales, de manière à ce qu'un maximum de solutions différentes restent après les deux étapes d'élimination. Les paramètres et techniques d'optimisation doivent être choisis avec un soin particulier de manière à réaliser cet objectif.

1.5.2 Optimisation globale

Paramètres valués

Notre approche a pour objectif de déterminer l'ensemble de ces variables de structure et de contrôle pour un manipulateur mobile. Les interactions existant entre ces différentes variables -entre la structure et sa commande par exemple- sont très complexes et ne permettent pas de réel découplage pendant la phase de conception préliminaire. Les paramètres à gérer pour ce travail sont en grand nombre, ils sont donnés dans le tableau 1.4.

STRUCTURE		CONFIGURATION	
Liaisons		Contraintes	
Nombre	ddl	Tâche	Réalisation
Type	R-P-F	Environnement	Obstacles
Axe	$\vec{x}_i, \vec{y}_i, \vec{z}_i$		
Paramètres	DH		
Eléments		Coordonnées	
Type	Barres, roues	Cartésiennes	(x_1, \dots, x_6)
Nombre	n_b, n_r	Opérationnelles	(q_1, \dots, q_{ddl})
Géométrie	longueur, rayon, dim. de la base	Efforts	Cartésiens Opérationnels

TAB. 1.4: Paramètres de conception et de configuration.

Chaque colonne de ce tableau correspond à une des phases de conception décrites au paragraphe précédent. Ainsi, la première phase d'élimination permettra d'extraire un ensemble de structures grossières -localement optimisées- dont la seule certitude réside dans l'aptitude à réaliser la tâche. Les paramètres qui en ressortent sont le nombre de degrés de liberté (ddl), le type de liaisons utilisées (rotoïde ou prismatique) et leurs axes respectifs. La géométrie des différents éléments mis en jeu est réellement optimisée au moment de la phase finale en parallèle avec la commande du robot. C'est aussi à ce moment-là que sont décidées les consignes à appliquer. L'évaluation des solutions proposées est fondamentale au cours de chaque étape d'optimisation, il est donc nécessaire d'apporter un soin tout particulier à l'établissement de la fonction d'évaluation.

Fonction objective multi-critères

L'évaluation des solutions robotiques dans la réalisation de la tâche est l'axe du travail de la conception optimisée. Celle-ci évolue avec les différentes phases de la conception présentées dans les paragraphes précédents. Seules les contraintes liées à la tâche sont invariantes, elles tiennent compte

- de l'accomplissement de la tâche : contrainte en position de l'organe terminal, en vitesse, en effort.
- de l'environnement : évitement d'obstacles.

Les autres critères d'évaluation sont liés à la *performance* de la solution dans la réalisation de la tâche. La fonction objective résultante regroupe donc un ensemble de critères très hétérogènes (binaires, contraintes fortes et contraintes "molles") que l'optimisation doit gérer. Cette non continuité de la fonction d'adaptation doit être prise en compte pour le choix des outils d'optimisation adaptés à notre problème.

Algorithmes évolutionnaires

Il est nécessaire d'utiliser un outil puissant pour l'optimisation qui réponde à nos exigences :

- optimisation globale sur un espace de recherche très vaste.
- Fonction objective irrégulière
- Peu d'information *a priori* sur le système.

La méthode d'optimisation par les algorithmes évolutionnaires présente toutes les qualités requises pour notre étude. Ils présentent l'avantage supplémentaire d'une approche par éliminations successives des solutions qui correspond à la méthodologie de conception que nous proposons, ainsi qu'un coût logiciel moyen pour la recherche. Ils nous permettent de constituer une "boîte noire" ayant en entrée la tâche à réaliser et ses contraintes et en sortie la solution structurelle optimale (constituée d'un assemblage de pièces mécaniques : actionneurs, barres, plate-forme...) et sa commande. Une manipulation aisée de ces assemblages est à considérer, surtout pour l'étape 1 de pré-sélection.

Approche modulaire

L'utilisation de modules présente trois avantages essentiels pour la conception de robots :

- Leur modélisation individuelle est simple puisque des pièces de géométries identiques sont manipulées, ce qui permet d'obtenir simplement le modèle général du robot considéré. Le nombre de modules différents doit rester faible pour complètement tirer avantage de ce point.
- Leur combinaison ouvre sur un (très) grand nombre de topologies différentes. Ce nombre augmente exponentiellement avec la quantité de modules utilisés. Cette diversité de structures permet l'abord de nombreuses tâches.
- L'adaptation de la structure à une nouvelle tâche éventuelle peut être envisagée dans une approche modulaire.

Notons de plus, d'un point de vue fabrication, que l'usinage de modules normés est plus accessible. Leur utilisation assure une plus grande robustesse des robots et une plus grande facilité de réparation lors d'une panne.

1.5.3 Conclusion

La problématique générale de notre étude a été posée dans ce premier chapitre. Notre travail propose une méthodologie basée sur les algorithmes évolutionnaires pour introduire la commande dans l'évaluation de la structure de robots manipulateurs mobiles lors de la phase de conception préliminaire.

La suite du mémoire est divisée en trois parties. Dans la première (chapitres deux et trois), l'approche évolutionnaire est décrite. Un algorithme est mis en place puis implémenté pour déterminer le contrôle basique et sur une tâche simple d'un manipulateur mobile constitué de robots de dynamique connue. Les problèmes liés à l'utilisation de l'outils évolutionnaire sont mis en évidence. Le rôle particulier de l'interprétation du

problème à résoudre y est étudié. L'analyse des réponses de l'algorithme d'optimisation permet alors de proposer des adaptations fiables et robustes vis-à-vis du cas étudié.

Le chapitre 4 présente la mise en place du module d'élimination préliminaire des structures des robots. Les structures réalisant des étapes ponctuelles de la tâche (sous-tâches la caractérisant) sont extraites parmi toutes les solutions étudiées. Cette recherche revient à effectuer une *étude de faisabilité* de la tâche par les solutions structurelles proposées par l'algorithme.

Le dernier chapitre met en œuvre la recherche des vecteurs d'entrées de commande de ces robots. La détermination des configurations initiales optimales du point de vue de la génération de consignes est mise en place. Les résultats, issus de la simulation, sont bien sûr discutés au cours de chaque étape ainsi que leurs limites respectives.

Chapitre 2

Approche par l'Optimisation Evolutionnaire

Les algorithmes évolutionnaires sont inspirés de la théorie de l'évolution de Darwin [Darwin59] et des lois génétiques introduites par Mendel [Mendel65]. Cette section expose l'historique et le fonctionnement général sur l'exemple de l'Algorithme Génétique Simple (AGS). Leurs avantages et inconvénients sont relevés à travers les différentes applications qui en ont déjà été faites et une première approche à notre problème est exposée et discutée, utilisant un algorithme évolutionnaire adapté à notre problème.

2.1 De l'évolution à l'évolutionnaire

2.1.1 Origines

Les Algorithmes Evolutionnaires sont une famille d'algorithmes basés sur l'association des lois génétiques et des règles de survie des espèces (Fig. 2.1).

Les lois génétiques ont été introduites par Mendel en 1865. Elles développent les caractéristiques des fonctions de reproduction, de croisement et de mutation des espèces sexuées à travers la manipulation de leurs gènes. L'existence d'une chaîne codée contenant les caractéristiques d'un individu y est déjà envisagée. L'évolution génétique à travers une génération résulte donc de modification de cette chaîne codée suivant des règles précises.

Les règles de survie des espèces à travers les générations ont été introduites par Darwin en 1859. Le paradigme Darwinien, "survie du mieux adapté", est axé sur quatre points essentiels : la sélection naturelle, la variabilité, l'objectif et l'adaptation résultante. La sélection naturelle avantage les espèces adaptées à leur environnement alors

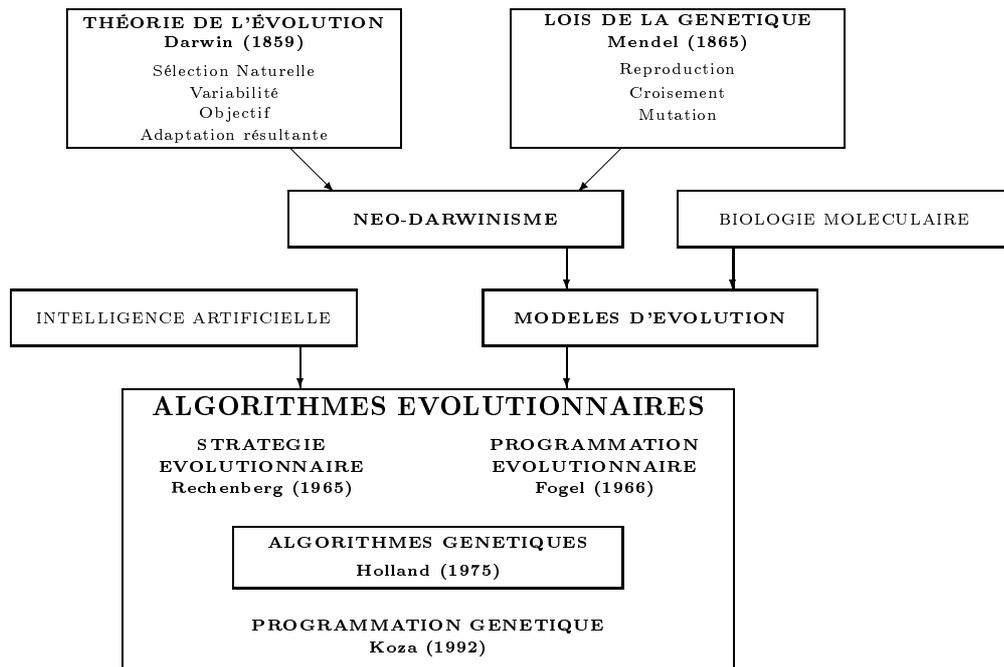


FIG. 2.1: Généalogie Evolutionnaire.

que la variabilité entraîne des petites variations aléatoires sur les parents qui se répercutent sur les enfants. Ces deux opérateurs mènent à la capacité de survivre et de se reproduire. L'adaptation des populations à leur environnement contribue aussi à la création de nouvelles espèces. Tous ces changements, aléatoires ou contraints, sont dirigés par le même objectif qui est la survie.

La figure 2.1 montre l'évolution historique qui a permis l'aboutissement des algorithmes évolutionnaires. Les influences dont ils découlent proviennent de domaines scientifiques très différents. En effet "association de ces deux travaux a donné naissance au *Néo-Darwinisme* au début du XX^{ème} siècle, puis à l'élaboration de modèles mathématiques de l'évolution par les biologistes. Des algorithmes d'optimisation nouveaux en ont été adaptés par les informaticiens. Les capacités toujours grandissantes en technologie informatique permettent aujourd'hui encore le développement de nouveaux algorithmes utilisant des fonctions de plus en plus complexes. Cependant ceux-ci peuvent encore être classés dans les trois grandes écoles fondatrices des algorithmes évolutionnaires qui se sont distinguées :

- Les stratégies d'évolution (Europe, Allemagne)
- La programmation évolutionnaire (ou évolutive, US côte ouest)
- Les algorithmes génétiques (US côte est)

Ces différentes approches sont décrites dans le paragraphe suivant.

2.1.2 Philosophies évolutionnaires

Si leurs philosophies diffèrent, les concepts de ces trois courants sont tous marqués du sceau darwinien. Les *Algorithmes Génétiques* sont de loin les plus connus parmi la famille évolutionnaire. Ils sont souvent considérés comme la base de ces algorithmes car ils en sont assez représentatifs. Issus de la recherche en intelligence artificielle et en biologie, ils ont été les premiers à être formalisés mathématiquement ([Holland75], Fig. 2.1) et surtout vulgarisés dans un ouvrage à grand succès [Goldberg89]. En réalité, les *Algorithmes Evolutionnaires* sont apparus dans les années 60 aux Etats-Unis.

Les *Stratégies Evolutionnaires* sont nées de la recherche appliquée en ingénierie mécanique et s'attachent surtout à faire du réglage de paramètres dans des problèmes de conception [Rechenberg65][Schwefel81].

La *Programmation Evolutive* s'est longtemps opposée aux algorithmes génétiques pour des raisons "philosophiques". Elle a été développée dans la recherche sur les automates à états finis ou les séries temporelles [Fogel66].

La *Programmation Génétique* est la dernière née des techniques évolutionnaires. Elle est largement et officiellement assimilée aux Algorithmes Génétiques. La méthode consiste à en appliquer les mécanismes à des programmes d'ordinateurs en passant par un formalisme de représentation en LISP (structure arborescente). Son idée d'associer cette représentation évolutionnaire a ouvert de nouvelles perspectives de recherche en intelligence artificielle [Koza92].

Toutes ces techniques utilisent en commun le vocabulaire de la biologie, de telle sorte que l'analogie complète existe entre biologie et optimisation (tableau 2.1). Ce rappro-

Biologie	Optimisation	Notation
Individu	Solution candidate	$X \in E$
Population	Sous-ensemble de Solutions	$P_i \in E$
Adaptation	Valeur de la fonction objective	$F(X)$
Génération	Itération de l'algorithme	$P_i \rightarrow P_{i+1}$
Génotype	Codage de la solution	$C(X)$ ou X
Phénotype	Solution décodée	$P(X)$ ou X

TAB. 2.1: Transposition sémantique du vocabulaire biologique

chement permet une compréhension plus aisée du fonctionnement de l'optimisation évolutionnaire dans le sens ou le parallèle avec l'évolution naturelle connue est facilement établi. Les différentes étapes représentatives de la méthodologie évolutionnaire sont décrites ci-après à travers une technique de base : l'Algorithme Génétique Simple.

2.2 Algorithme Génétique Simple (AGS)

Cet algorithme est le plus abordable et utilise d'une manière concise et directe tous les opérateurs fondamentaux des algorithmes évolutionnaires. L'AGS est représenté sur la figure 2.2 et est décrit dans [Goldberg89]. Nous allons clarifier les différentes étapes

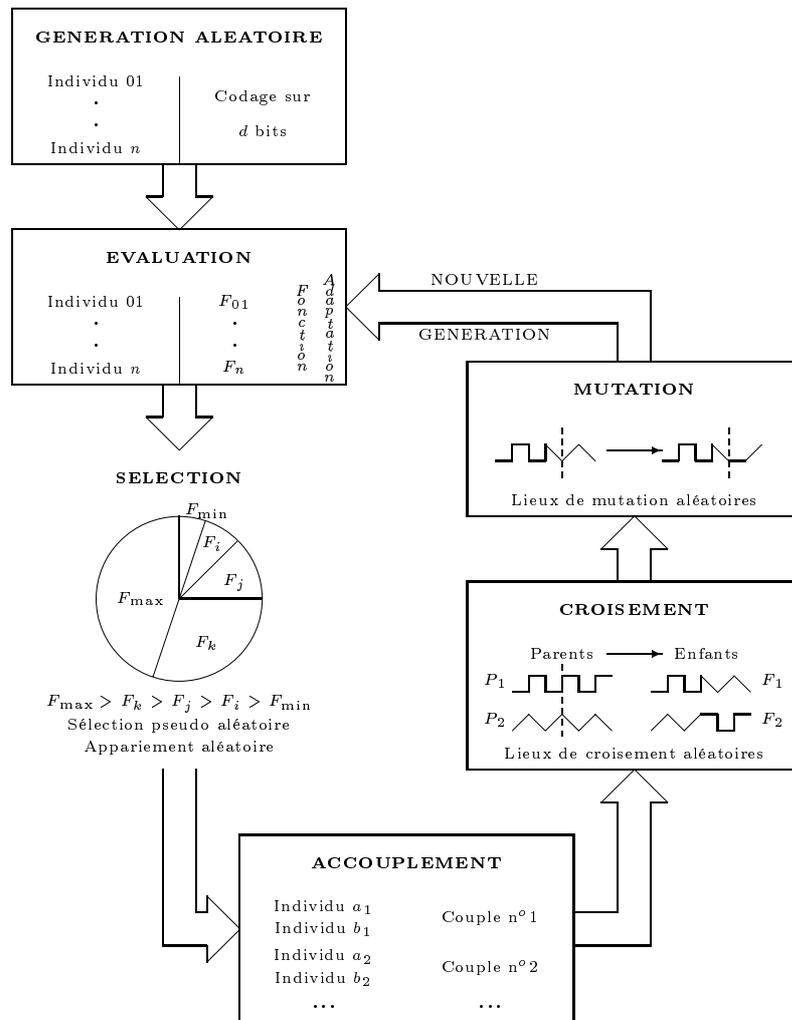


FIG. 2.2: Algorithme Génétique Simple (AGS).

de l'optimisation.

2.2.1 Initialisation

La première étape de la simulation de l'évolution génétique d'une population d'individus est d'en définir le point de départ. Celui-ci est déterminé dans l'initialisation de l'algorithme, où des individus sont générés aléatoirement selon une répartition uniforme sur l'espace de recherche. Ceux-ci sont des solutions potentielles au problème d'optimisation et constituent la population initiale. Chaque solution est codée dans

un mot binaire, appelé chromosome ou génotype, qui va être manipulé par les opérateurs génétiques. La première étape de l'AGS est donc de *définir un codage binaire des solutions recherchées* (traduction de réels en binaires, précision).

2.2.2 Evaluation

Chaque solution codée est ensuite évaluée par une fonction objective (ou d'adaptation, dite Fitness) qui regroupe tous les critères d'évaluation du système (contraintes, performances, ...). Ainsi une valeur d'adaptation qui représente sa performance est attribuée à chaque chromosome (chaîne binaire représentant l'individu).

2.2.3 Sélection

Une sélection est faite sur l'ensemble de la population initiale en extrayant une sous-population selon la méthode dite de la "roulette russe" [Goldberg89] : à chaque individu est attribué sur une roue un secteur d'angle proportionnel à sa valeur d'adaptation relative (Fig. 2.2). Par exemple, un individu dont l'adaptation est égale au quart de l'adaptation totale (somme de toutes les adaptations de la population) se verra attribuer un secteur équivalent à un quart de la roue. Ensuite, on fait tourner la roue autant de fois qu'il y a d'individus, ceux sélectionnés correspondant aux secteurs sur lequel la roue s'est arrêtée. Ces chromosomes sont rangés dans le pool des géniteurs. Cette méthode permet d'avoir un échantillonnage aléatoire biaisé de la population initiale selon un biais proportionnel à la valeur d'adaptation. L'espérance de reproduction d'un individu peut alors être définie comme le rapport de sa valeur d'adaptation par la valeur moyenne d'adaptation de la population. Cette méthode est également appelée sélection proportionnelle car l'espérance de reproduction est proportionnelle à l'adaptation. Lorsque la population des géniteurs est construite, les individus non sélectionnés sont oubliés. Les chromosomes géniteurs, eux, sont soumis aux opérateurs génétiques suivants.

2.2.4 Croisement à un point

Le premier opérateur qui suit la sélection est le croisement. Des couples de solutions sont formés pour émuler la reproduction sexuée. Deux parents (couple de chromosomes géniteurs) vont donner naissance à deux enfants par croisement, c'est-à-dire en échangeant un morceau de chaîne binaire. Le phénomène biologique associé est l'enjambement qui correspond à l'échange de morceaux d'ADN entre les chromosomes homologues d'une même paire lors d'une phase de la méiose. Les deux génotypes sont placés en vis-à-vis et un point de césure (point de croisement) est choisi dans la chaîne binaire. Les parties de chaîne situées après la césure sont échangées entre les deux chromosomes parents. La conséquence est la création de deux nouvelles chaînes binaires appelées chromosomes enfants. Ces derniers sont par définition complémentaires entre eux par rapport aux parents. Le croisement est appliqué à chaque couple de parents avec la probabilité de croisement P_c . Le principe du croisement est d'associer des

morceaux de solutions intéressants du point de vue de la représentation des caractéristiques recherchées afin de créer des solutions globales meilleures du point de vue de la performance dans la représentation de ces caractéristiques.

2.2.5 Mutation binaire

La mutation, du point de vue biologique, correspond au changement spontané d'une base d'acides aminés (ACGT) en une autre base dans la chaîne d'ADN. Elle peut être la conséquence de phénomènes extérieurs (rayonnement cosmique, radioactivité, ...) ou d'une erreur de recopie d'ADN. Dans l'AGS, cet opérateur change la valeur de chaque bit touché de la population enfant avec une probabilité de mutation P_m . Si la mutation est décidée, un 1 devient un 0 et inversement. La mutation permet d'explorer de nouvelles zones de l'espace de recherche en créant de nouvelles solutions dans la population.

2.2.6 Remplacement générationnel

Le remplacement de la population initiale par la nouvelle population obtenue après passage par les opérateurs s'appelle remplacement générationnel. Il ne laisse aucun espoir de survie aux parents en tant que tels. Le patrimoine génétique ne peut se transmettre qu'à travers les enfants. La nouvelle population subit à son tour les différentes étapes d'une génération pour laisser la place à une nouvelle génération. Le cycle continue jusqu'à satisfaction du critère d'évaluation.

2.3 Nature et limites de la méthode

2.3.1 Diversité des algorithmes

L'algorithme génétique simple précédemment décrit utilise les premiers opérateurs découverts et de nombreuses améliorations ont été apportées depuis. Les spécificités immuables de l'algorithme génétique sont le codage binaire et le croisement. En modifiant ces deux domaines, le spectre complet des algorithmes évolutionnaires connus ou inconnus peut être obtenu. La figure 2.3 montre le champ des combinaisons possibles pour réaliser un algorithme évolutionnaire connu.

Le codage peut subir des variations importantes, utiliser des nombres réels, entiers ou des structures de données plus complexes. Les opérateurs eux-mêmes peuvent supporter plusieurs modes d'action plus ou moins sophistiqués allant du simple réglage dynamique des probabilités jusqu'à l'optimisation globale de l'algorithme. La méthode la plus universelle consiste à utiliser un méta-algorithme qui va lui-même évoluer à travers ses opérateurs en incluant sa stratégie d'optimisation dans les paramètres de la solution. Ainsi, la recherche va traiter parallèlement le problème et l'algorithme. Cependant le temps nécessaire est généralement très long et la solution optimale n'est pas garantie, puisque l'algorithme final n'est pas représentatif de l'évolution totale de la solution.

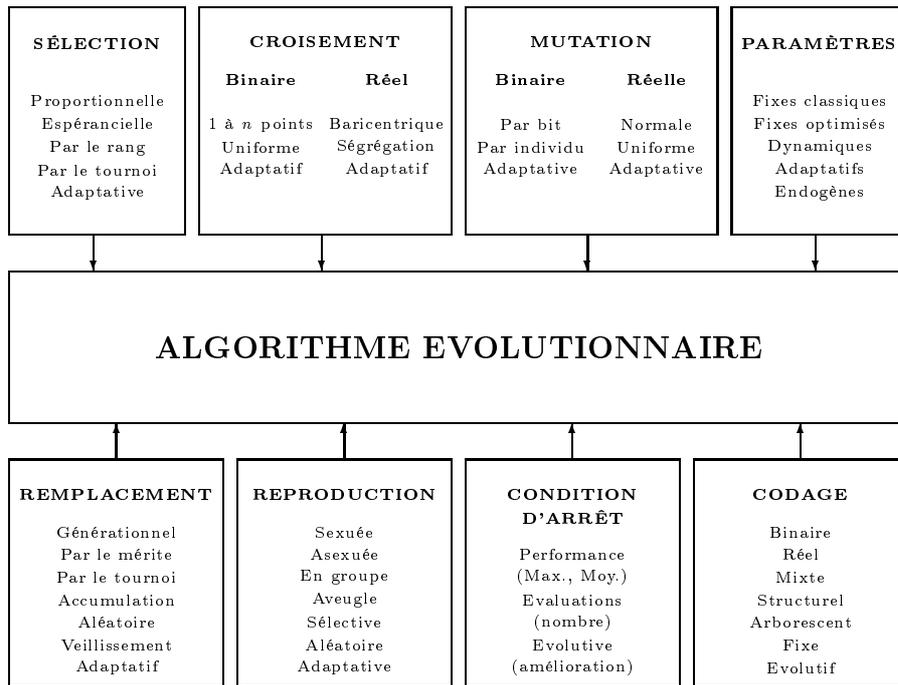


FIG. 2.3: Diversité des algorithmes évolutionnaires.

2.3.2 Avantages et inconvénients

La méthode évolutionnaire possède de nombreux atouts. Nous en avons déjà souligné certains dans le chapitre 1 que les méthodes déterministes ne possèdent pas. Les avantages et inconvénients peuvent être résumés ainsi :

Avantages

1. Codage des paramètres (déplacement des minima locaux)
2. Travail sur une population (parallélisme)
3. Utilise un modèle “boite noire” (pas d’informations auxiliaires)
4. Règles stochastiques (permet d’explorer l’espace)

Inconvénients

1. Travail d’adaptation du problème (codage)
2. Optimum global non garanti (suboptimal)
3. Réglage des paramètres (opérateurs)

Le travail d’adaptation du problème concerne le codage/décodage des solutions et surtout la fonction d’évaluation de ces solutions. En effet, les manipulations génétiques vont être appliquées à la solution codée. Il faudra donc interpréter et évaluer correctement ces modifications quelles qu’elles soient. De plus, si l’algorithme converge,

la solution évoluée n'est pas nécessairement l'optimum global. En effet, l'utilisation d'opérateurs stochastiques ne prévient pas d'une solution meilleure, même proche du meilleur individu découvert.

Enfin, plusieurs phénomènes peuvent apparaître dans un algorithme évolutionnaire. Certains sont désirables, d'autres sont néfastes à la recherche. D'autres sont à doser finement selon le problème à résoudre :

Phénomènes désirables

1. Convergence partielle
2. Découverte de nouvelles solutions
3. Mémoire des régions intéressantes

Phénomènes indésirables

1. Convergence prématurée
2. Egarement de la recherche
3. Disparition des bonnes solutions

Phénomènes à doser

1. Convergence
2. Exploration
3. Exploitation

Tous ces phénomènes sont influencés par les paramètres appliqués aux différents opérateurs génétiques (sélection, mutation, croisement). Il est donc nécessaire de porter une attention toute particulière à ces paramètres afin d'en découvrir le meilleur réglage possible. Plusieurs tentatives ont été réalisées pour mettre en place des méthodes de détermination des paramètres optimaux [Grefenstette86] [Sebag97]. Malheureusement, seuls les ordres de grandeur sont invariants car il s'avère que chaque problème d'optimisation a ses propres particularités et que l'on doit reconsidérer en général ces paramètres à chaque nouveau problème.

2.3.3 Convergence et théorème des schémas

Les résultats relatifs à la convergence des algorithmes évolutionnaires sont peu nombreux et assez limités. En effet, comme nous l'avons déjà vu, l'utilisation des opérateurs stochastiques engendre beaucoup d'incertitudes. Les seuls résultats de convergence obtenus pour les algorithmes évolutionnaires sont incomplets : la convergence n'est démontrée que pour des cas réducteurs des algorithmes [Rudolf97] [Cerf94]. Le théorème des schémas proposé dans [Holland75] est la première tentative de modélisation mathématique du comportement des algorithmes évolutionnaires. Les schémas sont des

sous-espaces génotypiques regroupant des individus qui possèdent certains allèles (valeurs de bit) en commun.

Holland définit la valeur d'adaptation d'un schéma comme étant l'adaptation de tous les individus qui l'incluent dans la population courante. Le théorème donne l'espérance du nombre de représentants d'un schéma d'une génération à l'autre en fonction des paramètres d'évolution :

$$E[N(H, t + 1)] \geq N(H, t) * \frac{F(H, t)}{\tilde{F}(\Omega, t)} * [1 - P_c * \frac{\ell(H)}{N - 1} - P_m * o(H)]$$

$N(H, t)$:	Nombre de représentants du schéma H à la génération t
$F(H, t)$:	Adaptation du schéma H calculé dans la génération t
$\tilde{F}(\Omega, t)$:	Adaptation moyenne de la génération t
P_c	:	Probabilité de croisement
P_m	:	Probabilité de mutation
N	:	Nombre de bits dans un génotype
$\ell(H)$:	Longueur du schéma H (écart entre le premier et le dernier bit)
$o(H)$:	Ordre du schéma H (nombre de bits instanciés)

Ce théorème conduit à ce qu'il est convenu d'appeler le dogme génétique : *les schémas courts et performants (adaptés) augmentent dans la population*. Ces schémas sont appelés les briques élémentaires et seraient donc au cœur de la dynamique évolutionnaire. Le théorème implique également que l'alphabet binaire est le plus performant car il possède une cardinalité minimale. Cette théorie a été abondamment critiquée dans la littérature pour plusieurs raisons :

- le parallèle entre schéma optimum et solution optimale n'est pas si évident [Goldberg89],
- la cardinalité de l'alphabet n'est pas pertinente [Antonisse89],
- la performance d'un schéma dépend fortement de sa variance dans la population [Radcliffe95].

Depuis, cette théorie a été remplacée par d'autres plus rigoureuses et moins controversées comme la théorie des *formae* [Radcliffe95]. Il apparaît cependant que l'application reste spécifique au problème traité [Soper00] et l'analyse de convergence doit être traitée au cas par cas.

2.3.4 Dilemme exploration/exploitation

Le phénomène le plus prédominant dans une optimisation évolutionnaire est le fragile équilibre qu'il existe entre l'exploration et l'exploitation. L'exploration consiste à aller chercher des nouvelles solutions là où elles n'ont pas encore été cherchées. Cette caractéristique est fortement favorisée dans les algorithmes de type Monte-Carlo ou exhaustif. L'exploitation consiste à rester le plus proche possible de la meilleure solution déjà rencontrée. Ce principe de conservation est fortement utilisé dans les méthodes déterministes comme le gradient ou le Hill-Climbing. Une bonne méthode d'optimisation

devra doser ces deux ingrédients avec un soin tout particulier. En effet, la meilleure piste de recherche peut très bien se situer toute proche de la meilleure solution trouvée comme très éloignée.

Dans la méthode évolutionnaire, le dosage exploration-exploitation est pris en charge par la pression de sélection (exercée par l'opérateur de sélection) et par la pression de variation (exercée par les opérateurs de mutation et de croisement). Dans la sélection, la pression détermine le taux de conservation des meilleures souches génétiques. Dans la mutation et le croisement, c'est le taux de variation des individus géniteurs qui est contrôlé. Cette méthode permet de doser au mieux les deux composantes de la recherche, d'une manière dynamique (en fonction de la population) ou même adaptative.

2.3.5 Pré-requis pour l'optimisation évolutionnaire

Avant de réaliser efficacement un algorithme évolutionnaire, il faut satisfaire plusieurs exigences. Nous pouvons classer les pré-requis pour l'optimisation évolutionnaire en deux catégories (tableau 2.2) :

Pré-requis	
de pertinence	de réalisation
Grands espaces de recherche	Formalisation pour le codage des solutions
Paramètres de nature hétérogènes	Procédure d'évaluation automatique et robuste
Irrégularités de la fonction objective	Travail d'adaptation et de mise au point

TAB. 2.2: Les deux catégories de pré-requis pour l'optimisation évolutionnaire

- Les *pré-requis de pertinence* regroupent toutes les particularités de l'optimisation qui font que l'approche évolutionnaire est intéressante à appliquer.
- Les *pré-requis de réalisation* rassemblent toutes les conditions nécessaires liées à l'application du calcul évolutionnaire au problème traité.

Les pré-requis de pertinence sont présents, comme nous l'avons vu dans le chapitre précédent, dans notre problème d'optimisation combinatoire. Les pré-requis de réalisation sont accessibles dans l'état actuel de la conception en robotique. Leur intérêt a été montré sur des problèmes plus ou moins simples de conception. Les différents problèmes soulevés sont appliqués et détaillés dans la section suivante. Le problème que nous y traitons est une étude simplifiée du problème global qui nous intéresse et vise à mettre en évidence la pertinence des techniques évolutionnaires pour notre approche.

2.4 Suivi de trajectoire

Nous nous proposons, dans cette partie, d'utiliser la technologie évolutionnaire face au problème de la commande d'un manipulateur mobile. Pour cela, des robots connus

sont représentés par leurs modèles dynamiques. Une tâche simple est mise en place et les consignes à appliquer sur chaque actionneur pour la réaliser sont recherchées. L'objectif de l'étude, les robots utilisés et la tâche sont décrits ci-après. Les différentes étapes d'utilisation des algorithmes évolutionnaires sont retracées par la suite, puis les résultats discutés en fin de section.

2.4.1 Position du problème

L'objectif

L'objectif de cette étude est d'analyser le comportement des algorithmes évolutionnaires pour l'optimisation. Nous proposons de rechercher un jeu de consignes (entrées de commande sur chaque actionneur du robot) qui permette de suivre une trajectoire, définie d'un point de départ X_i à un point d'arrivée X_d donnés. Plusieurs hypothèses sont formulées afin de cadrer la recherche :

- Une tâche simple est déterminée, de manière à pouvoir vérifier la validité des solutions proposées et la qualité des améliorations apportées.
- Une simulation réaliste est mise en place. Les robots (plate-forme mobile et manipulateur) sont représentés par leurs modèles dynamiques respectifs.
- La recherche s'effectue sur un seul jeu de consignes, c'est-à-dire que le manipulateur mobile doit s'acquitter au mieux de la tâche avec une consigne constante sur chaque actionneur.

Cette recherche revient donc à effectuer une commande en boucle ouverte sur le robot, avec un retour d'information sur la réalisation de la tâche en fin de simulation. Ainsi, si l'algorithme parvient à déterminer le jeu de consignes, nous pourrions envisager des tâches plus complexes. Il faut de plus déterminer les limites de validité de la recherche de solutions de manière à s'assurer l'adaptabilité de l'approche sur d'autres systèmes.

Les robots

Le manipulateur mobile que nous utilisons est constitué par l'assemblage de deux robots existants et représentés, pour la simulation, par leurs modèles dynamiques :

- le robot RoMAIN, plate-forme mobile non-holonome possédant deux roues motrices à différentiel de vitesse à l'arrière et deux roues directrices à l'avant. Cette plate-forme a été modélisée et sa commande dynamique étudiée dans [Ruau98].
- d'un manipulateur PUMA 560 à six axes dont le modèle dynamique peut être obtenu dans [PUMA560].

Les caractéristiques dynamiques de ces deux robots sont rappelées en annexe 2. Leurs modèles ont été validés par l'expérimentation à travers un travail de thèse portant sur sa conception pour RoMAIN [Ruau98] et de nombreux documents sont disponibles sur le PUMA 560 (ce manipulateur a en l'occurrence fait l'objet d'une étude par les algorithmes génétiques utilisés pour la résolution du problème géométrique inverse dans [Chocron00]).

La tâche

Une tâche simple de suivi de trajectoire par l'effecteur du manipulateur mobile est demandée. Elle consiste à suivre une ligne droite horizontale sur 5 mètres le long d'un mur (figure 2.4). L'orientation de l'organe terminal n'est pas prise en compte, c'est-à-dire que la mobilité du poignet du manipulateur n'est pas utilisée. Dans cette partie

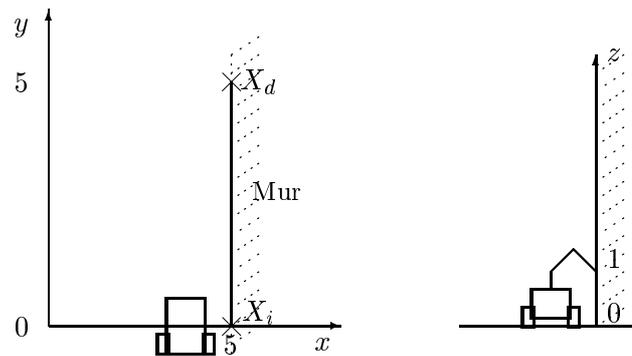


FIG. 2.4: Représentation de la trajectoire et configuration initiale du manipulateur mobile

de l'étude, le robot ne dispose que d'un seul jeu de consignes pour réaliser cette tâche. L'erreur maximale autorisée sur la position de l'effecteur est de 5 millimètres, c'est-à-dire que la tâche sera validée à 5 millimètres près du point de vue de l'évaluation. Les configurations finales ou intermédiaires du manipulateur mobile ne sont pas contraintes. La configuration initiale pose la plate-forme mobile parallèlement au mur et dans la direction de la trajectoire désirée, comme représenté sur la figure 2.4. La base du bras, placée au centre de la plate-forme mobile, est à la même ordonnée que l'effecteur. La situation initiale de celui-ci coïncide avec le point de départ X_i de la trajectoire (figure 2.4). Le tableau 2.3 donne les valeurs des coordonnées généralisées du bras manipulateur ainsi que la position et orientation de la plate-forme mobile à l'instant initial. L'effecteur est placé en $X_i = (5, 0, 1)$ et doit atteindre $X_f = (5; 5; 1)$ (figure

Paramètres configurationnels	Plate-forme			Manipulateur		
	x_b	y_b	θ	ϕ_1	ϕ_2	ϕ_3
Valeurs initiales	4.5 m	0 m	$\pi/2$ rd	1.89 rd	0.72 rd	0 rd
Valeurs finales attendues	4.5 m	5 m	$\pi/2$ rd	1.89 rd	0.72 rd	0 rd

TAB. 2.3: Configuration initiale des robots et prévision finale

2.4). Une solution évidente apparaît immédiatement pour atteindre sans problème la position finale désirée, puisqu'il suffit que le véhicule avance parallèlement au mur, c'est-à-dire d'avoir une consigne nulle sur les actionneurs du manipulateur et deux consignes égales sur les roues de la plate-forme. Nous obtiendrions alors les résultats présentés sur la seconde ligne du tableau 2.3. Cette évidence a ses avantages : elle permet de valider directement la pertinence physique des individus, c'est-à-dire toute

la méthodologie décrite précédemment. Les solutions proposées par l'algorithme sur cette première approche basique sont étudiées de manière à tester leur validité, leur fiabilité et leur robustesse.

Conclusion

Cette étude est destinée à tester le comportement de l'algorithme et améliorer ses performances. Bien que simple, la tâche à réaliser est très exigeante compte tenu de la précision demandée et de la distance à parcourir. Nous sommes confrontés immédiatement à plusieurs problèmes :

- Les solutions recherchées (consignes) ne peuvent pas être évaluées directement, mais par l'intermédiaire d'un retour sur le comportement du robot (avancée de l'effecteur sur la trajectoire, précision du suivi).
- La forte redondance du système (d'ordre 3) ainsi que sa sensibilité aux faibles variations sont des obstacles supplémentaires à la détermination des consignes.
- La simulation doit laisser le temps au robot d'affiner une bonne trajectoire potentielle. Cela limite les possibilités d'élimination trop rapide des solutions et est donc gourmand en temps de calcul.

Un compromis est donc à déterminer entre la rigueur de la sélection et le temps de calcul, amplifié par l'évaluation indirecte des solutions.

La mise en place de l'algorithme s'articule de la manière suivante : la première étape consiste à énoncer clairement le problème d'optimisation et définir les variables sans ambiguïté. Elle débouche sur l'établissement du *chromosome* représentant la solution, établit sa précision et la dimension de l'espace de recherche. La seconde étape décrit les critères pour l'évaluation ainsi que la fonction coût globale et sa précision. Enfin la détermination des opérateurs évolutionnaires et des paramètres associés achèvent l'établissement de l'algorithme d'optimisation général.

2.4.2 Algorithme évolutionnaire

Enoncé du problème d'optimisation

Les vecteurs qui interviennent dans le problème sont les suivants :

Vecteur des consignes	:	$U = (u_1, u_2, u_3, u_4, u_5)^t$
Configuration	:	$(x_b, y_b, \theta, \phi_1, \phi_2, \phi_3)^t$
Coordonnées généralisées	:	$q = (\phi_d, \phi_g, \phi_1, \phi_2, \phi_3)^t$
Coordonnées opérationnelles	:	$X = (x_e, y_e, z_e)^t$

Le vecteur des consignes regroupe les valeurs des 5 entrées de commande destinées aux actionneurs (figure 2.5). Le vecteur configuration contient la position et orientation de la plate-forme mobile, ainsi que les 3 valeurs articulaires du manipulateur. Il y a équivalence directe entre ce vecteur et celui des coordonnées généralisées, en utilisant les relations de contraintes de la plate-forme mobile posées en annexe 3. Le vecteur des coordonnées opérationnelles regroupe l'abscisse, l'ordonnée et la cote de l'organe terminal

du manipulateur mobile dans le repère de référence. Les relations liant ces différents vecteurs sont obtenues par les modèles dynamiques des deux robots et discutées plus loin. Le problème que nous nous proposons de résoudre peut alors s'énoncer comme un problème d'optimisation basique :

$$\begin{aligned} \text{Trouver} \quad & \mathbf{U}^* = U(\ddot{\mathbf{q}}^*) \in \mathbf{R}^5 \\ \text{tel que} \quad & F(\mathbf{X}^*) \leq F(\mathbf{X}) \quad \forall \mathbf{X} \in \mathbf{R}^3 \end{aligned}$$

avec \mathbf{q} Vecteur des coordonnées généralisées
 \mathbf{X} Vecteur des coordonnées opérationnelles
 U Vecteur des consignes sur les actionneurs
 $F(\mathbf{X})$ Fonction d'évaluation des entrées de commande

La relation entre la solution optimale recherchée \mathbf{U}^* et les variables utilisées pour son évaluation \mathbf{X}^* est donnée par les modèles dynamiques du manipulateur et de la plate-forme mobile.

Représentation des solutions

La définition claire des variables à optimiser est la première étape de l'étude. Dans notre cas, on recherche les consignes de couple des roues arrières (motrices à différentiel de vitesse) de la base mobile et des trois axes du manipulateur. Le vecteur des coordonnées généralisées s'écrit

$$\begin{aligned} \mathbf{q} &= (q_1, q_2, q_3, q_4, q_5) \\ &= (\phi_d, \phi_g, \phi_1, \phi_2, \phi_3) \end{aligned}$$

où ϕ_d et ϕ_g sont les valeurs articulaires des roues droite et gauche respectivement, ϕ_1 , ϕ_2 et ϕ_3 celles de la hanche, de l'épaule et du coude du manipulateur (figure 2.5). On doit donc déterminer un vecteur d'entrées de commande à 5 inconnues, tel que

$$\mathbf{U} = (u_1, u_2, u_3, u_4, u_5) \text{ avec } u_i \in \mathbf{R}_{[0, b_i]}$$

où u_i représente la consigne à appliquer à l'actionneur i et b_i la borne supérieure qui lui est liée, c'est-à-dire le couple maximal applicable à l'articulation i (tableau 2.4). Les consignes sont envoyées sur les articulations correspondantes. Nous commencerons par adimensionner les paramètres d'optimisation. Cette procédure classique dans les problèmes d'optimisation multivariés est conseillée dans l'évolution artificielle car ces paramètres doivent être manipulés par des opérateurs différents. Il est important de les quantifier objectivement, nous utiliserons donc le vecteur $\tilde{\mathbf{U}}$ des entrées de commande adimensionnées :

$$\tilde{\mathbf{U}} = (\tilde{u}_1, \tilde{u}_2, \tilde{u}_3, \tilde{u}_4, \tilde{u}_5) \text{ avec } \tilde{u}_i = \frac{u_i}{b_i} \in \mathbf{R}_{[0,1]}$$

\tilde{u}_i représente alors la valeur adimensionnée liée à la consigne u_i appliquée sur l'arti-

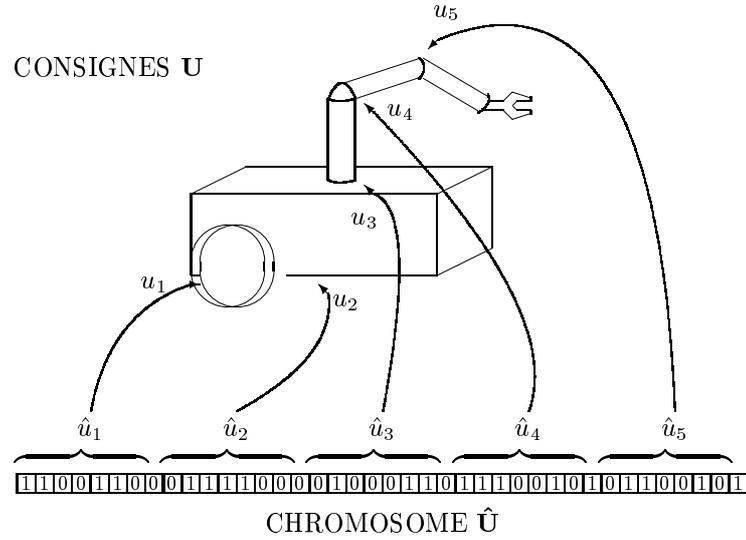


FIG. 2.5: Représentation du chromosome lié au vecteur des consignes

culatation i . Le *chromosome* est constitué par le regroupement des 5 mots binaires \hat{u}_i de longueur ℓ , représentant les 5 variables \tilde{u}_i :

$$\hat{\mathbf{U}} = \langle \hat{u}_1 \quad \hat{u}_2 \quad \hat{u}_3 \quad \hat{u}_4 \quad \hat{u}_5 \rangle \quad \text{avec} \quad \tilde{u}_i = \frac{\text{Val}(\hat{u}_i)}{(2^\ell - 1)}$$

Ce chromosome résultant de l'association des 5 mots binaires est illustré à la figure 2.5. Les informations codées qu'il contient constituent le phénotype de l'individu (\hat{u}_i). Les informations décodées correspondent à son génotype (u_i). Nous définissons un gène comme la plus petite entité de codage (0 ou 1 dans notre cas). Les mots binaires sont codés arbitrairement sur 8 bits, ce qui donne un chromosome binaire de $5 * 8 = 40$ bits. Ce choix implique une précision moyenne sur les solutions. Il se justifie par les difficultés de convergence qui apparaissent très rapidement avec la longueur des chromosomes. La précision δ_i sur la valeur d'une variable u_i vaut

$$\delta_i = \frac{b_i}{2^\ell - 1}$$

où $\ell = 8$ dans notre cas. δ_i représente l'erreur de discrétisation commise sur la consigne appliquée à l'articulation i . Nous obtenons donc, en utilisant les caractéristiques technologiques des robots, les valeurs rapportées dans le tableau 2.4. La dimension de l'espace de recherche est

$$\Omega = 2^{5*\ell} = 1.10^{12}$$

La très grande dimension de cet espace est, comme nous l'avons vu, l'une des justifications de l'utilisation d'un algorithme génétique pour la recherche. Une autre justification se trouve dans l'expression de la fonction d'évaluation des solutions.

Actionneur	Couples max b_i	Erreurs δ_i
Roue gauche	6.25 Nm	0.024 Nm
Roue droite	6.25 Nm	0.024 Nm
Hanche	9.9 Nm	0.039 Nm
Epaule	14.9 Nm	0.058 Nm
Coude	9.1 Nm	0.036 Nm

TAB. 2.4: Couples max et erreur de discrétisation sur les consignes pour chaque articulation et pour un codage sur 8 bits

Evaluation des solutions

L'évaluation est une partie délicate car elle ne se fait pas directement sur les variables qui évoluent. Les critères doivent rester simple tout en donnant une valeur de performance fiable sur l'état du système. Les modèles des robots permettent d'établir les effets des consignes proposées sur leur comportement et ainsi d'évaluer la solution. Cette approche est donc indépendante du robot considéré puisque la commande s'effectue en boucle ouverte. Elle ne s'intéresse qu'aux entrées (n_b consignes) et sorties (situation) du système représenté. La seule donnée essentielle est finalement le nombre de ddl de la structure qui permet de dimensionner le chromosome.

Trois critères sont pris en compte pour évaluer une solution potentielle. Ils sont fixés de manière à effectuer une évaluation objective des solutions, c'est-à-dire orientés vers la tâche (réalisation et performance) et non sur l'intuition. Les deux premiers concernent donc la réalisation de la tâche : l'avancement sur la trajectoire et la précision de positionnement. Le troisième est l'évitement de collision avec l'obstacle que constitue le mur.

Relation consignes/position de l'effecteur : le suivi est assuré par la prise en compte de l'environnement et l'erreur en position. La performance de la solution est basée sur la valeur de cette erreur, et la distance parcourue le long de la trajectoire désirée. Le modèle des robots permet d'établir le lien entre la consigne \mathbf{U} et le vecteur des variables opérationnelles $\mathbf{X} = (x_e; y_e; z_e)$. On peut écrire, de manière générale :

$$\begin{aligned}\ddot{\mathbf{q}}(t) &= M^{-1}(\mathbf{U} - B) \\ \dot{\mathbf{q}}(t) &= \int \ddot{\mathbf{q}} dt\end{aligned}$$

- $\mathbf{q}(t)$ vecteur des coordonnées généralisées
- $\dot{\mathbf{q}}(t)$ vecteur des vitesses généralisées
- $\ddot{\mathbf{q}}(t)$ vecteur des accélérations généralisées
- $M(q)$ matrice d'inertie du robot considéré
- $B(q, \dot{q})$ vecteur regroupant les effets inertiels, centrifuges et gravitationnels

Dans le cas du manipulateur, la position de l'organe terminal dans la base liée au bras est obtenue directement par le modèle géométrique via l'intégration des vitesses articulaires :

$$\mathbf{q}_m(t) = \int_t^{t+\delta t} \dot{\mathbf{q}}_m dt$$

$$\mathbf{X}_m(t) = f(\mathbf{q}_m)$$

\mathbf{q}_m vecteurs des coordonnées généralisées du manipulateur (ϕ_1, ϕ_2, ϕ_3)

\mathbf{X}_m Position de l'effecteur dans la base liée à la plate-forme mobile

Dans le cas de la plate-forme mobile, le vecteur des vitesses opérationnelles est déduit du modèle cinétique puis intégré pour obtenir sa position dans le repère de référence (repère fixe lié à l'environnement) :

$$\dot{\mathbf{X}}_b(t) = J(q)\dot{\mathbf{q}}_b$$

$$\mathbf{X}_b(t) = \int_t^{t+\delta t} \dot{\mathbf{X}}_b dt$$

\mathbf{q}_b vecteurs des coordonnées généralisées de la plate-forme mobile (ϕ_d, ϕ_g)

\mathbf{X}_{base} position (x_b, y_b) et orientation θ dans le repère de référence

$J(q)$ jacobienne des vitesses de la plate-forme mobile

La position de l'organe terminal dans le repère de référence est donc

$$\mathbf{X}(t) = \mathbf{X}_b(t) + \mathcal{R}_{b/0}\mathbf{X}_m(t)$$

où $\mathcal{R}_{b/0}$ représente la matrice de rotation de la base autour de l'axe vertical :

$$\mathcal{R}_{b/0} = \begin{pmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Distance parcourue : Dans les limites de notre étude, seule l'évolution de l'ordonnée opérationnelle nous intéresse (figure 2.4). Les notations utilisées ici peuvent s'appliquer sans aucun problème au cas général. Soient X_i la position de départ et X_d la position finale désirée. La distance parcourue est estimée par le critère

$$C_1 = \frac{\|X_f - X_d\|}{\|X_i - X_d\|}$$

où X_f représente la position atteinte par l'effecteur à la fin de la simulation. $C_1(X)$ est une courbe majorée par 1 lorsque $X_f = X_i$ et minorée par 0 lorsque $X_f = X_d$. Ce critère est à minimiser : il traduit la proportion de trajectoire restant à parcourir pour atteindre X_d .

Erreur en position : L'erreur en position ε de l'effecteur est l'écart entre la position effective et la position désirée de l'effecteur. A l'instant t , elle s'écrit

$$\varepsilon(t) = \|\mathbf{X}_d(t) - \mathbf{X}(t)\|$$

où $\mathbf{X}_d(t)$ est le vecteur position désirée de l'organe terminal à l'instant t et $\mathbf{X}(t)$ son vecteur opérationnel. Dans notre cas, nous désirons que l'effecteur suive une trajectoire à abscisse et côte constantes. Donc $\varepsilon(t)$ concerne l'écart en x et z à la trajectoire, soit

$$\varepsilon(t) = \sqrt{(x(t) - x_d)^2 + (z(t) - z_d)^2}$$

où $x(t)$ et $z(t)$ représentent respectivement l'abscisse et l'altitude de l'effecteur à l'instant t , $x_d = 5$ et $z_d = 1$ l'abscisse et l'altitude de la trajectoire désirée. le critère d'erreur adimensionné entre la trajectoire désirée et la position effective de l'organe terminal à l'instant t est évalué par le rapport entre l'erreur finale atteinte à la fin du déplacement ε_f et l'erreur maximale autorisée ε_ℓ :

$$C_2 = \frac{\varepsilon_f}{\varepsilon_\ell}$$

Le seuil ε_ℓ permet de limiter le temps de simulation alloué à une solution trop éloignée de l'objectif. La valeur de ε_ℓ est délicate à fixer puisqu'elle est une des conditions d'arrêt de la commande en cours. En effet, si l'effecteur s'écarte trop de la trajectoire, la simulation est stoppée et considère l'individu suivant. Si ε_ℓ est trop petit, C_2 tend à un critère binaire et les solutions trouvées relèveront essentiellement de l'aléatoire. Si cette valeur est trop grande, le temps de simulation s'en ressent. Nous fixons cette valeur à l'écart existant entre la base du bras et le mur à l'instant initial, soit 50 cm. Cette valeur permet de trouver dès le départ un ensemble de consignes qui entraînent le robot dans la bonne direction.

Ce critère est majoré par 1 pour $\varepsilon_f = \varepsilon_\ell$ et minoré par 0 lorsque $\varepsilon_f = 0$. On peut prévoir que l'erreur finale atteinte aura la valeur de l'erreur limite en fin de simulation. Ainsi, C_2 atteindra sa valeur maximale (1) la plupart du temps. L'influence de ce critère prendra effet essentiellement lorsque la distance à parcourir sera réalisée. Les très bonnes solutions se distingueront alors par l'écart final à la trajectoire.

Obstacles : Un critère booléen est utilisé, pénalisant grandement la configuration qui présente une intersection entre une partie du robot et le mur, seul obstacle présent dans notre simulation. Pour ce faire, des points de surveillance sont placés autour du robot (Fig. 2.6). Le mur représente alors une surface délimitant une zone interdite pour chacun de ces points. La figure 2.6 montre deux exemples de situations pour le robot. Le premier représente un cas sans collision. Les points de surveillance y sont notés de 1 à 6. La plate-forme mobile est représentée par une enveloppe parallépipédique l'incluant totalement. Quatre points de surveillance sont placés à chaque angle de cette enveloppe. Deux supplémentaires sont à l'épaule et au coude du manipulateur. Le deuxième cas

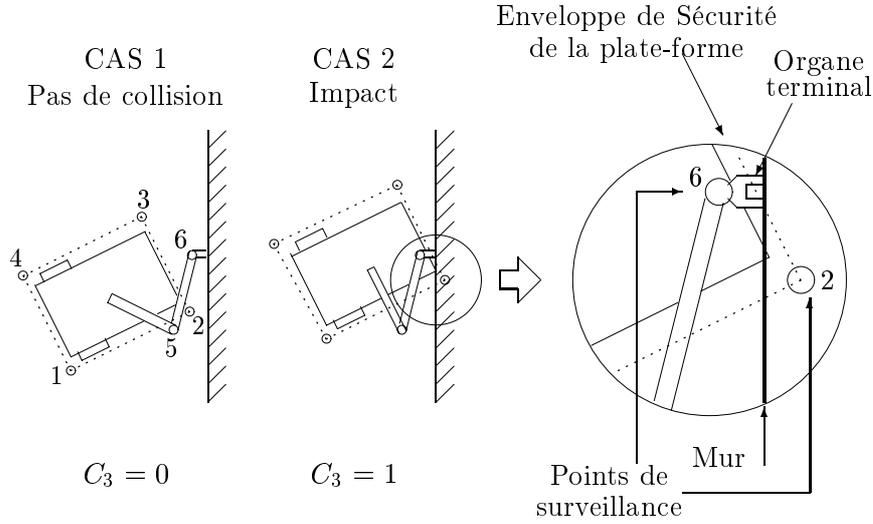


FIG. 2.6: Evitement de collision avec le mur

de la figure 2.6 considère une collision. Le point de surveillance 2 dépasse la limite autorisée. Le critère d'évitement de collision avec le mur s'écrit donc

$$C_3 = \begin{cases} 1 & \text{si un des points de surveillance entre dans la zone interdite} \\ 0 & \text{sinon} \end{cases}$$

Ce critère élimine automatiquement les cas de collision car la pénalité est très élevée.

Fonction objective résultante : Les trois critères définissent un demi-cylindre, de rayon ε_l et limitée par le mur, autour de la trajectoire à parcourir. La fonction exponentielle e^{-x} est utilisée pour construire la fonction d'évaluation finale. Celle-ci permet d'affiner l'évaluation des solutions car elle est plus sensible pour des valeurs de x comprises entre 0 et 1 (Fig. 1.7) : la croissance plus lente de la fonction exponentielle lorsque x est supérieur à 0.5 présente l'avantage de permettre une sélection large des solutions au départ de l'optimisation. Sa croissance rapide lorsque x tend vers 0 permet de différencier les très bonnes solutions. La fonction coût s'écrit donc

$$F = e^{-\frac{k_1 * C_1 + k_2 * C_2 + k_3 * C_3}{k_1 + k_2 + k_3}}$$

Où les k_i sont les poids, constants, liés aux critères respectifs C_i .

La somme pondérée des trois critères est bornée par 0 et 1, ce qui correspond à une valeur de F entre 0,35 et 1. La valeur maximale de F est atteinte pour une valeur minimale des critères. Une solution parfaite aura donc 1 comme valeur d'évaluation, correspondant à l'atteinte de la position finale désirée avec une erreur de position de l'effecteur nulle et aucune collision avec le mur. Pour la simulation, aucun critère n'est favorisé devant les autres puisque tous concernent de manière égale la réalisation de

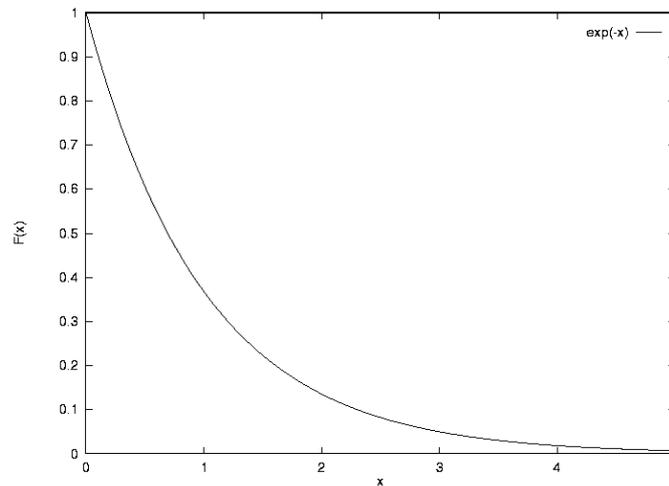


FIG. 2.7: Evolution de la fonction e^{-x} autour de 0.

la tâche : les trois gains k_1 , k_2 et k_3 sont fixés à 1. Le caractère binaire du critère d'évitement d'obstacles assure sa réalisation. Les deux autres critères devront être minimisés.

Cette fonction est donc à maximiser et son optimisation se fait à F_ε près, où F_ε représente l'erreur résultant de la discrétisation des variables. L'erreur de consigne estimée au paragraphe précédent et les modèles directs des deux robots nous permettent de calculer l'erreur maximale sur la position de l'effecteur due à la discrétisation des consignes :

$$\varepsilon_e = 2 * 0.2 * \delta_1 + 0.8 * \delta_3 + 0.2 * \delta_4 + 0.8 * \delta_5 = 0.081 \text{ m}$$

$$F_\varepsilon = \frac{1}{3} * 2 * \frac{\varepsilon_e}{46.40} = 0.012$$

Cette estimation doit être effectuée de manière à obtenir la condition d'arrêt de l'algorithme (la meilleure évaluation est recherchée à F_ε près). Il est à remarquer ici que l'erreur due à la discrétisation des variables est de 8 cm, ce qui est énorme. L'utilisation de génotypes codants sur 10 bits, par exemple, permettrait de réduire cette erreur à 2 cm. De l'augmentation résultante de l'espace de recherche (de l'ordre de 10^{15} au lieu de 10^{12}) résulterait une augmentation similaire du temps de calcul pour la simulation. L'utilisation d'un génotype binaire se justifie dans notre étude par la possibilité d'utiliser des relations mathématiques établies pour la détermination des paramètres génétiques, comme nous le verrons par la suite.

Opérateurs évolutionnaires

Des opérateurs très avancés sont actuellement disponibles dans la littérature [Sebag97] [Schoenaur98]. Les opérateurs de sélection, mutation et croisement que nous utilisons ici sont cependant sensiblement les mêmes que ceux de l'AGS (Fig. 2.2). Les modifications apportées tendent à conserver l'aspect générique de leur utilisation (pas d'adaptations spécifiques au problème). Elles sont issues de la littérature et adoptées pour des raisons d'efficacité reconnues.

L'opérateur de sélection se nomme la roulette avec reste stochastique [Goldberg89]. Elle permet de conserver automatiquement les individus dont la valeur d'adaptation est supérieure à la moyenne en laissant aux autres individus une chance d'être sélectionnés dans le groupe des géniteurs. Elle est basée sur la partie entière de l'espérance de recopie $E(x_i)$ accordée à chaque individu x_i ($i = 1..k$) où k est la taille de la population :

$$E(x_i) = \frac{F(x_i)}{\sum_{i=1}^k F(x_i)} * k = \frac{F(x_i)}{\tilde{f}}$$

$F(x_i)$: valeur d'adaptation de la $i^{\text{ème}}$ solution candidate
 $\tilde{f} = \frac{1}{k} \sum_{i=1}^k F(x_i)$: valeur d'adaptation moyenne de la population
 k : taille de la population

Le restant est réparti entre les individus candidats selon la méthode de l'AGS (roue proportionnelle).

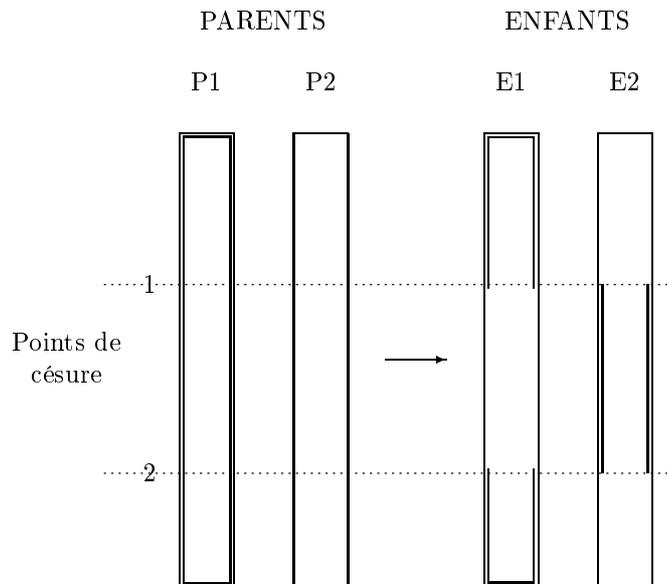


FIG. 2.8: Principe du croisement à deux points de césure

L'opérateur de croisement est basé sur celui de l'AGS. Il utilise cependant deux points de césure au lieu d'un (figure 2.8). La partie du chromosome échangée n'est plus limitée au début et à la fin mais peut être la partie centrale du chromosome. La probabilité de croisement dans un couple est fixée par le paramètre de croisement p_c discuté plus loin. L'opérateur de mutation est rigoureusement identique à celui de l'AGS. Il effectue une mutation binaire avec une probabilité de mutation p_m .

Paramètres évolutionnaires

Les paramètres que nous devons fixer sont les probabilités de croisement p_c et de mutation p_m , le nombre de générations g ainsi que la taille de la population k . L'évolution est peu sensible à la variation de la probabilité de croisement, une valeur de 50% est adoptée, valeur généralement admise dans la littérature évolutionnaire [Goldberg89]. La détermination de p_m est plus fondamentale puisque ce paramètre détermine la proportion d'aléatoire dans l'évolution (une valeur de 50% correspond à une recherche purement aléatoire). Les applications évolutionnaires précédentes [Goldberg89] montrent que l'équilibre entre mutation et croisement présente un bon compromis pour la dynamique évolutionnaire. Cela permet de fixer la valeur de la probabilité de mutation :

$$p_m = \frac{1}{2 * n_b * \ell} = 0.0125$$

Chaque individu a alors une chance sur deux de ressortir inchangé du processus de mutation.

Le produit des paramètres k et g détermine le nombre maximum d'évaluations qui vont être accordées à l'optimisation évolutionnaire. La première contrainte à respecter est bien sûr le temps de calcul du processus d'optimisation. Comme nous l'avons dit plus haut, un des avantages majeurs de la méthode évolutionnaire est qu'elle n'évalue qu'une portion infinitésimale de l'espace de recherche. Cette contrainte d'infinitésimalité s'écrit

$$\frac{k * g}{\Omega(E)} \ll 1 \quad \text{avec} \quad \Omega(E) = 2^{n_b * \ell} = 2^{40} = 1.10^{12}$$

où E représente l'espace des solutions candidates et $\Omega(E)$ son cardinal. Le rapport entre k et g reste alors à déterminer.

La taille minimale de la population est fixée par l'erreur fonctionnelle de discrétisation F_ϵ

$$k_{\min} = 1 + \frac{1}{F_\epsilon} = 86.69$$

Cette valeur minimale nous permet d'avoir une bonne distribution des valeurs de la fonction objective. Avec $k = 100$, nous avons théoriquement accès à toutes les valeurs d'adaptation.

Le nombre de génération minimal, quant à lui, est fixé par

$$g_{\min} = n_b * \ell = 40$$

Cette valeur minimale permet d'assurer un processus d'évolution à proprement parler. Il est basé sur le temps nécessaire pour qu'un chromosome ait le temps d'être transformé en son complémentaire par la seule mutation avec une espérance du nombre de mutation égal à 1 (chaque bit a une chance sur $n_b * \ell$ d'être touché par la mutation). Ce nombre se doit d'être majoré car, comme nous le verrons par la suite, la simulation montre que l'évolution n'est pas encore terminée au bout des 40 générations. [Chocron00] propose de calculer une valeur optimale du nombre de générations, qui maximise la probabilité que chacun des $n_b * \ell$ bits du chromosome soient mutés une fois et une seule, soit :

$$\max \{P(n_b * \ell = 1)\} = C_{g_{opt}}^{n_b * \ell} * \left(\frac{1}{n_b * \ell}\right)^{n_b * \ell} * \left(1 - \frac{1}{n_b * \ell}\right)^{g_{opt} - n_b * \ell}$$

Appliquée à notre problème, cette relation donne $g_{opt} = 1642$ générations. Cette valeur est elle-même à remettre en cause car les changements n'apparaissent plus après la 800^{ème} génération. Ce paramètre est donc fixé en fonction du comportement de l'algorithme et de manière à ne pas handicaper son évolution complète et l'émergence d'une solution adaptée. Nous choisissons donc la valeur intermédiaire $g = 300$ générations. L'ensemble des paramètres est résumé dans le tableau 2.5. Les conditions ultérieures

Paramètre	Symbole	Valeur
Probabilité de croisement	p_c	0.5
Probabilité de mutation	p_m	0.0125
Taille de la population	k	100
Nombre de générations	g	300

TAB. 2.5: Paramètres d'évolution

d'utilisation de l'algorithme imposent en effet une convergence rapide (inclusion de la recherche des consignes dans le choix d'une structure optimale). Les modifications que nous apporterons par la suite s'orientent toutes dans ce sens. La condition d'infinésimale de l'espace

$$\frac{k * g}{\Omega(E)} = \frac{100 * 300}{1.10^{12}} = 3.10^{-8}$$

est bien vérifiée.

Algorithme général

L'algorithme général utilisé pour la simulation est présenté sur la figure 2.9. Il résume les différents aspects du problème que nous avons traités successivement dans cette partie. Chaque module, représenté dans un cadre, est indépendant : les k individus de la population y sont traités simultanément. La condition d'arrêt est basée sur

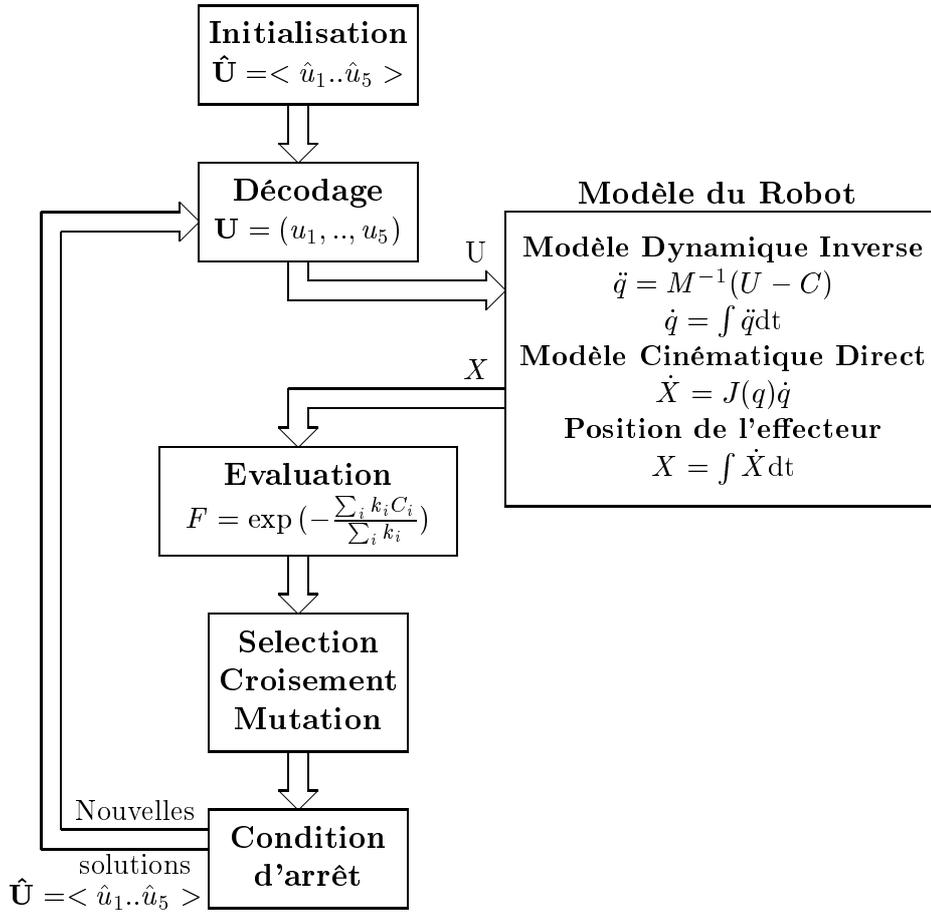


FIG. 2.9: Algorithme pour la simulation

l'obtention d'une valeur d'adaptation maximale (à F_ϵ près) ou lorsque le nombre maximal des générations est atteint. Nous allons maintenant présenter les résultats quant au comportement de l'algorithme et aux solutions extraites.

2.4.3 Premiers résultat de simulation

Convergence de l'algorithme

L'analyse du comportement de l'algorithme passe essentiellement par l'étude de l'évolution de la fonction d'évaluation des candidats avec les générations. De manière à tester statistiquement sa fiabilité, la simulation est relancée 100 fois (comprenant initialisation de la population), et les résultats obtenus moyennés sur ces 100 évolutions. Le graphe d'évolution représenté sur la figure 2.10 résulte d'un moyennage de la valeur d'adaptation F sur 100 évolutions différentes et sur 300 générations. Son allure générale met en évidence la justesse du choix des paramètres évolutionnaires par une croissance continue jusqu'à la convergence. La courbe supérieure suit l'évolution du meilleur individu pour chaque génération, tandis que la courbe inférieure représente la

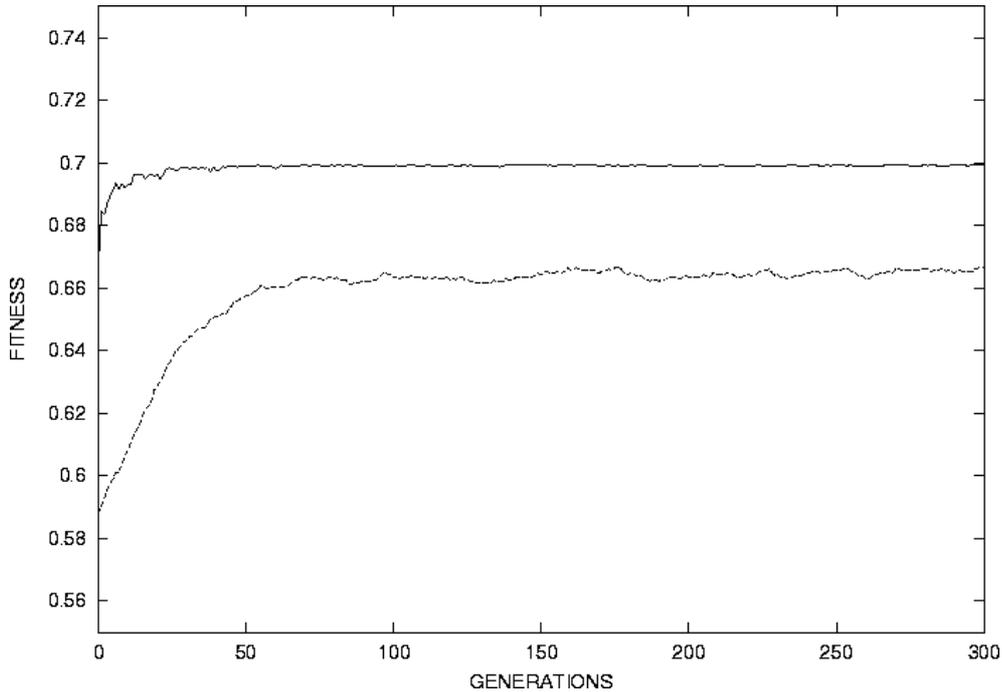


FIG. 2.10: Evolution de la fonction d'adaptation au cours des générations (— Meilleur - - - Moyenne)

valeur d'adaptation pour l'ensemble de la population. Il est important que ces deux courbes aient la même allure générale. Il en ressort une part raisonnable d'aléatoire dans la recherche de solution. La convergence vers la valeur d'adaptation maximale est très rapide et l'évolution se stabilise dès la vingt-cinquième génération pour le meilleur individu et vers la cinquantième génération pour la moyenne de la population.

La solution qui nous intéresse ne fait pas partie des meilleurs individus puisque la valeur maximale de la fonction coût ($F = 1$) n'est pas atteinte. La valeur maximale en est même très éloignée puisque se situant à 0,7 environ (celle-ci correspond approximativement à une somme non pondérée des trois critères de 1,1). La valeur moyenne de la population se stabilise à 0.66. Notons de plus la faible amplitude de l'évolution, puisque, pour le meilleur individu, celle-ci part de 0,67 pour se stabiliser à 0,7. La validité physique des solutions proposées est discutée ci-après, ainsi que le comportement correspondant du robot. Ces résultats nous permettront d'apporter des modifications allant vers l'amélioration des performances des individus.

Le temps mis pour une simulation complète, sur 300 générations, est d'environ 2 heures 40 minutes. Il varie en fonction de la qualité des solutions trouvées dès le départ. En appliquant le couple maximal sur les deux roues de la plate-forme mobile, le véhicule peut couvrir la distance de 5 mètres en 3'06" (des couples moyens de 3 N.m permettent de réaliser cette même distance en 4'42"). Chaque individu testé se voit donc alloué un temps maximal de 5 minutes pour évaluer la commande. Ainsi, si aucune vitesse

minimale de réalisation n'est demandée explicitement, les comportements trop lents se verront sanctionner faute de parvenir à l'objectif. Leur évaluation, cependant, restera dans une bonne moyenne du fait de la faible valeur du critère d'erreur en position.

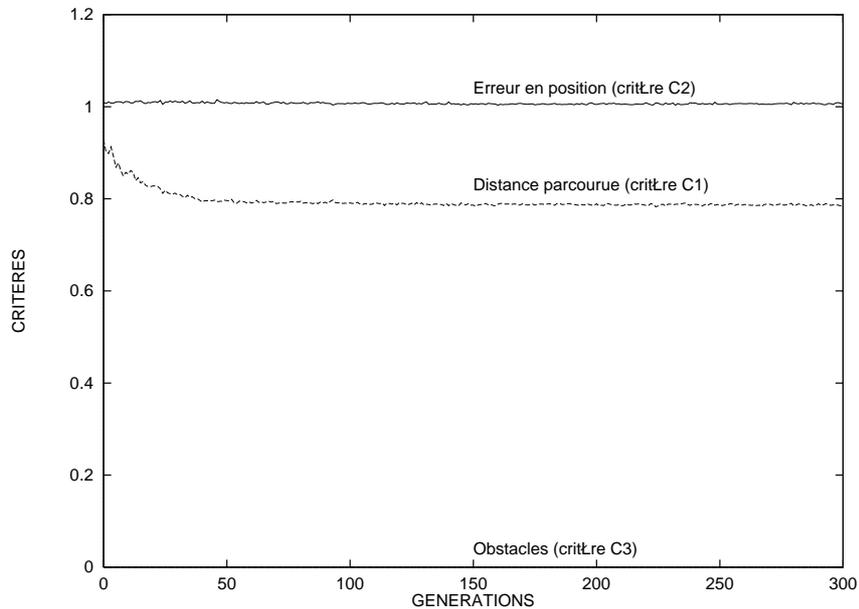


FIG. 2.11: Evolution de la valeur des critères des meilleurs individus en fonction des générations

Le deuxième graphique (figure 2.11) regroupe de manière similaire l'évolution des valeurs de chaque critère en fonction des générations. Ces courbes montrent que l'évaluation repose essentiellement sur la trajectoire parcourue. Comme prévu, le critère lié à l'évitement d'obstacles, C_3 , est maintenu à zéro. Par contre, nous remarquons que seul évolue le critère C_2 lié à l'avancement de l'organe terminal sur la trajectoire. Le critère C_1 destiné à minimiser l'écart à la trajectoire désirée reste à 1, ce qui rend son efficacité transparente du point de vue de l'évolution. En effet, celui-ci ne peut intervenir tant que l'organe terminal du manipulateur mobile n'est pas parvenu à hauteur de la position finale désirée.

Les valeurs des 5 consignes sont très variables d'une génération à l'autre. L'algorithme trouve beaucoup d'individus pour faire évoluer l'effecteur dans une zone s'étalant jusqu'à 1m50 vers la position finale désirée dans une marge d'erreur de 50cm de la trajectoire désirée (!). Cependant, les valeurs proposées mettent une fois de plus en évidence la non obtention du résultat attendu. Au vu de ces résultats, la première conclusion penche vers le manque d'efficacité de la recherche évolutionnaire. Afin d'apporter les améliorations nécessaires, une étude plus poussée sur l'influence des différents paramètres est effectuée et présentée ci-après. Comme nous l'avons remarqué précédemment en se basant sur l'allure générale des courbes d'évolution, les opérateurs évolutionnaires ne sont pas directement mis en cause. Leur comportement, classique,

doit cependant être adapté pour agir en synergie avec le problème traité. De manière à mettre en évidence la source du problème de détermination de meilleures solutions, l'influence des paramètres qui fixent les dimensions de la recherche est analysée par la suite (borne supérieure de l'erreur, distance à parcourir, nombre de degrés de liberté, redondance géométrique, paramètres évolutionnaires).

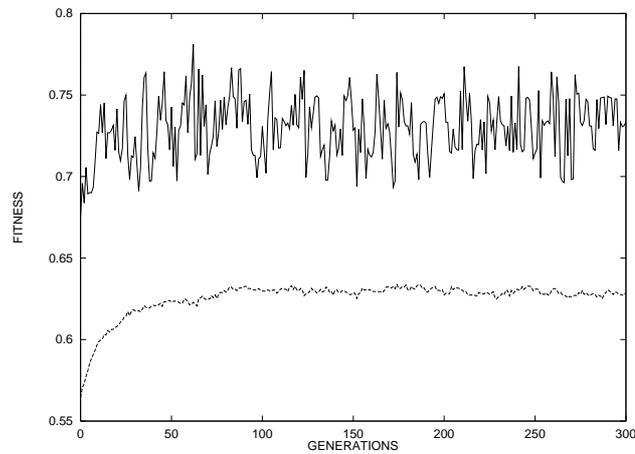
Influence des paramètres de la recherche

Dans cette partie de l'étude, nous présentons la meilleure solution issue d'une évolution complète de 300 générations. La convergence de l'algorithme présentée ci-dessus permet de ne pas remettre en cause directement les opérateurs évolutionnaires. L'absence de bonnes solutions trouve donc sa source dans la manière dont ces opérateurs interprètent le problème que nous posons. De manière à apporter à l'algorithme les améliorations adaptées, l'influence des variables liées à la pose du problème est étudiée :

- la définition des contraintes liées à la tâche (borne supérieure ε_ℓ de l'erreur en position de l'effecteur, distance à parcourir et caractéristiques) ;
- dimensions des solutions (nombre de ddl, redondance) ;
- influence des différents paramètres évolutionnaires.

(a) Contraintes de tâche La recherche évolutionnaire fonctionne par éliminations successives. L'écart maximal autorisé à la trajectoire avant élimination a une influence directe sur cette recherche dans le sens où une valeur plus élevée de la tolérance à l'erreur maximale favorise l'émergence de nombreux individus, même si leur évaluation n'est pas concluante du point de vue de la tâche. Le graphe 2.12 montre les évolutions des performances des solutions proposées (c'est-à-dire l'évolution de la valeur de C_1) au cours des générations en fonction de ε_ℓ . Afin d'obtenir une comparaison objective de ces résultats, le critère C_2 reste adimensionné par la valeur d'origine de ε_ℓ , c'est-à-dire 0,5 m. L'influence de la valeur limite n'atteint pas la totalité des évolutions. Elle se traduit plutôt à travers les possibilités d'émergence de bonnes solutions. En effet, lorsque la limite autorisée est fixée à 5 cm, c'est-à-dire que la simulation d'une solution amenant l'effecteur à plus de 5 cm est automatiquement stoppée, aucune solution, même moyenne, n'émerge de l'évolution. A partir de 20 cm et au fur et à mesure que cette valeur augmente, des solutions allant dans le sens positif de l'évaluation apparaissent plus nombreuses. Si cette limite ne permet pas l'émergence systématique, elle l'encourage et doit donc être considérée. La valeur intermédiaire de 0,5 cm semble un bon compromis entre l'émergence de solutions encourageantes et le temps allouable pour tester une solution.

Une deuxième remarque sur ce critère d'erreur C_2 concerne son influence sur la sélection. En choisissant la fonction la plus simple, c'est-à-dire en se concentrant sur la position finale atteinte par l'effecteur au bout de la simulation, on obtient des situations délicates dès le début de l'évolution, comme celle illustrée par la figure 2.13. Ces deux trajectoires représentent l'éloignement de l'effecteur à la trajectoire désirée en fonction



(a) Evolution de la fonction d'adaptation

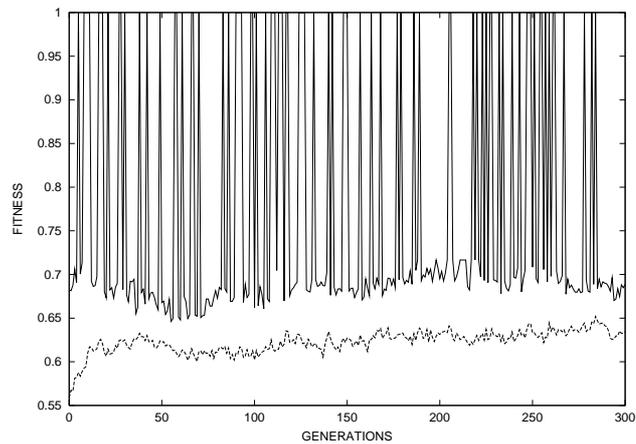
(b) Evolution de la valeur du critère d'avancement C_1

FIG. 2.12: Comparaison des évolutions avec variation de la tolérance à l'écart à la trajectoire ε_ℓ (— Meilleur - - - Moyenne).

de l'avancement sur cette trajectoire. Elles sont considérées comme équivalentes du point de vue du critère d'erreur. Or il est évident que la trajectoire 1 (figure 2.13) ne permettra jamais d'atteindre la cible 5 mètres plus loin et devrait être immédiatement éliminée de la sélection. Ce critère doit donc être modifié de manière à être plus actif dans la sélection d'une part (le fait que sa valeur soit stabilisée à 1 n'aide en rien l'évolution), d'autre part pour lever les éventuelles ambiguïtés de trajectoires comme celle rencontrée ici.

(b) Nombre de ddl et redondance géométrique Dans cette partie, la mobilité de la hanche du manipulateur est supprimée et sa consigne forcée à zéro, ce qui enlève

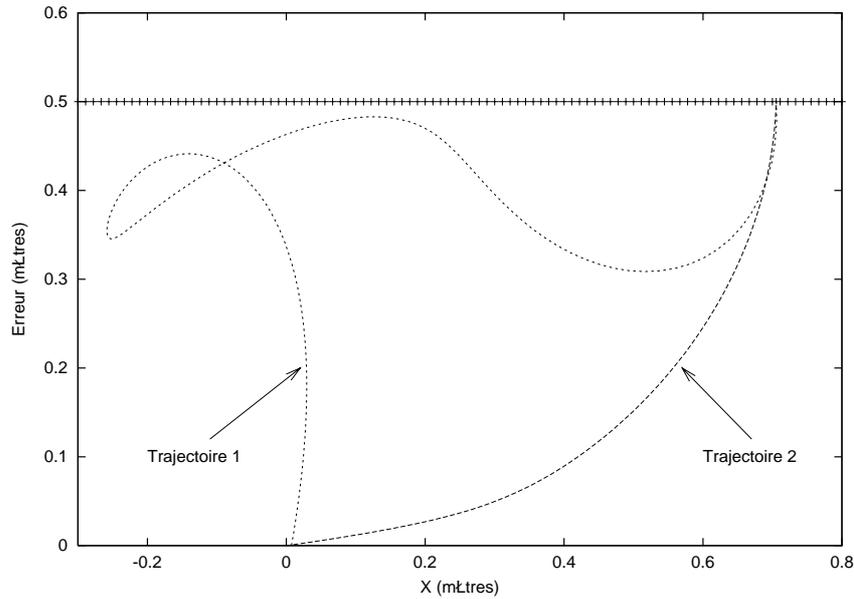
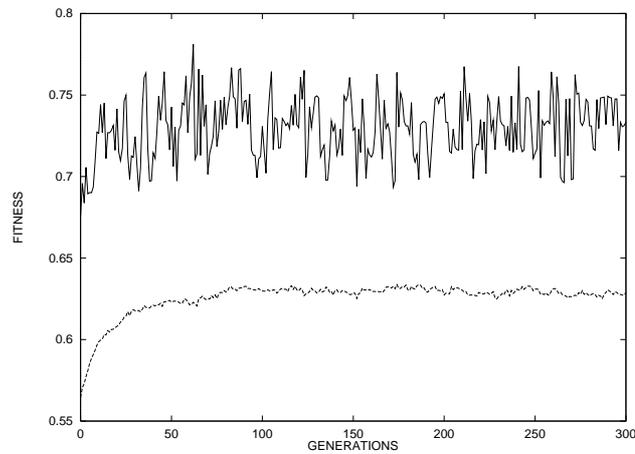


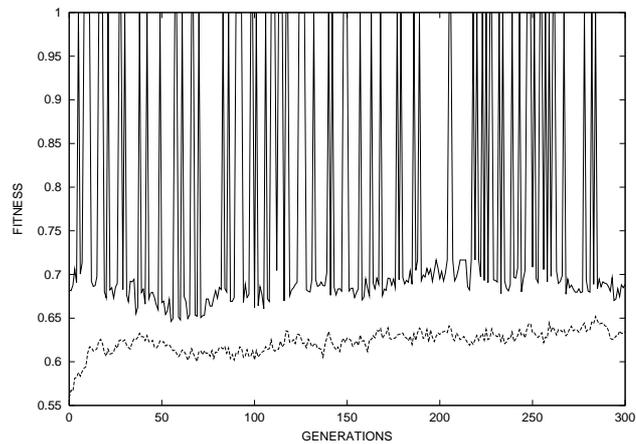
FIG. 2.13: Deux trajectoires équivalentes du point de vue du critère d'erreur C_2 .

un degré de redondance à la structure (rotation autour de l'axe vertical). La recherche s'effectue dans les mêmes conditions et l'objectif est toujours de déterminer les quatre consignes restantes pour réaliser le suivi. Le paramètre de mutation p_m est adapté et fixé à 1,0156 et les autres paramètres restent inchangés. Les graphes d'évolution correspondant à cette recherche sont représentés sur la figure 2.14. Le graphe de gauche (figure 2.14(a)) résulte d'un moyennage sur 100 évolutions et montre un caractère très dentelé de la valeur d'adaptation pour le meilleur individu d'une génération à l'autre. En comparaison avec les résultats précédents, les valeurs ont un ordre de grandeur comparable (figure 2.10). Par contre, l'évolution est totalement instable du point de vue des meilleurs individus. La figure 2.14(b) représente l'évolution de cette valeur sur une seule évolution, qui fournit l'explication de l'allure du moyennage de la figure 2.14(a). L'allure irrégulière montre que l'aspect exploitation de la recherche est très négligé devant l'exploration. Les solutions très proches sont peu envisagées. Ceci provient d'une valeur trop élevée de la probabilité de mutation. Nous pouvons remarquer que l'algorithme parvient à trouver des solutions intéressantes qui permettent d'atteindre la cible, contrairement au cas de la recherche des 5 consignes pour lequel aucune solution n'émergeait. En effet, des pics apparaissent à $F = 1$, que la perte d'information d'une génération à l'autre fait disparaître. Ceci a pour conséquence une scission de la courbe du meilleur individu en deux courbes distinctes entre les très bonnes solutions (l'effecteur parvient à la cible) et les autres (l'effecteur s'éloigne très rapidement de la trajectoire). L'algorithme trouve donc, ou ne trouve pas, de solution. Les solutions intermédiaires, quant à elles, apparaissent peu sur ces graphes.

La courbe 2.14(b) est obtenue dans 62% des cas seulement, et dans tous ces cas une excellente solution a émergé avant la douzième génération. Cette observation nous



(a) Moyenne sur 100 simulations



(b) Résultat d'une simulation

FIG. 2.14: Evolution de la fonction d'évaluation en fonction des générations (— Meilleur - - - Moyenne)

amène à considérer l'importance de l'initialisation sur toute la recherche évolutionnaire suivante, ainsi -et surtout- celle de l'apport de nouveaux individus tout en conservant le patrimoine génétique acquis. Il est évident, au vu de ces résultats, que la redondance cinématique du système étudié gêne la recherche et empêche l'algorithme d'évoluer. Ainsi, la pose du problème évolutionnaire doit tenir compte du couplage qui existe entre les variables à optimiser. Ceci nous amène à observer l'influence des paramètres évolutionnaires sur notre recherche.

(c) Paramètres évolutionnaires Des modifications de la taille de la population k ou de la probabilité de croisement p_c ont peu d'incidences sur l'évolution de la po-

pulation. La probabilité de mutation p_m , en revanche, joue un rôle non négligeable. La figure 2.15 reproduit les schémas d'évolution pour différentes valeurs de p_m (0.005, 0.0125, 0.05, 0.1 et 0.15).

Ces graphes montrent des "mauvaises" évolutions dans le sens où de bien meilleures solutions sont plus souvent proposées dans les mêmes conditions. Le cas $p_m = 0.0125$, par exemple, est la plus mauvaise évolution fournie (stagnation complète de l'évolution, non représentative de la moyenne présentée sur la figure 2.10). Si ce cas ne peut pas être considéré comme représentatif des évolutions, il ne peut pas non plus être considéré comme exceptionnel car il arrive dans 18% des simulations. Dans le problème de la prise en compte de la commande dans la boucle d'optimisation de la topologie d'un robot, nous ne serons pas en mesure de choisir l'évolution intéressante. Il nous faut donc envisager les pires situations, surtout lorsque leur apparition n'est pas accidentelle.

Les courbes font ressortir un nombre de bonnes solutions croissant avec la valeur de p_m . d'un autre côté, l'augmentation de cette valeur est aussi liée à l'aspect cahotique des résultats d'une génération à l'autre. La valeur de 0.53, qui apparaît comme un palier pour la valeur d'adaptation des meilleurs individus pour tous les exemples de la figure 2.15, correspond à deux critères restant à 1 dans la fonction d'évaluation. Ce sont donc des solutions de collision immédiate avec le mur ($C_1 = C_3 = 1, C_2 = 0$) ou d'éloignement très rapide de la trajectoire ($C_1 = C_2 = 1, C_3 = 0$). Ces courbes font aussi ressortir l'égarement de la recherche dans les cas présentés et la non exploitation (la perte) des bonnes solutions trouvées. L'observation de ces évolutions isolées nous amènent de plus à mettre en doute la validité d'une analyse du comportement de la recherche évolutionnaire sur les seuls moyennages. Par la suite, l'étude sera effectuée sur une seule évolution représentative des cas les moins bons.

2.5 Conclusions

Les conclusions pour cette première étude, bien que globalement négatives du point de vue des résultats obtenus, permettent de bien comprendre le comportement de l'optimisation génétique. Les problèmes auxquels nous avons été confronté ne sont pas exceptionnels mais peu relevés dans la littérature. A ces difficultés "classiques" (essentiellement liées à la pose du problème) s'ajoutent celles, particulières, liées au problème traité.

2.5.1 Egarement de la recherche

La faible amplitude de l'évolution des valeurs de la fonction d'adaptation montrent la non adaptation de l'algorithme évolutionnaire face à notre problème de commande, tout au moins dans le cadre que nous avons défini. Cette stagnation dans l'évolution trouve sa source dans

- le caractère couplé des variables à déterminer. Pour améliorer une solution qui éloigne la plate-forme mobile de la trajectoire par exemple, la solution s'oriente

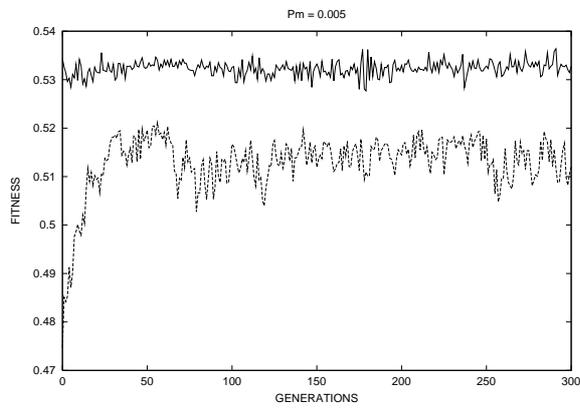
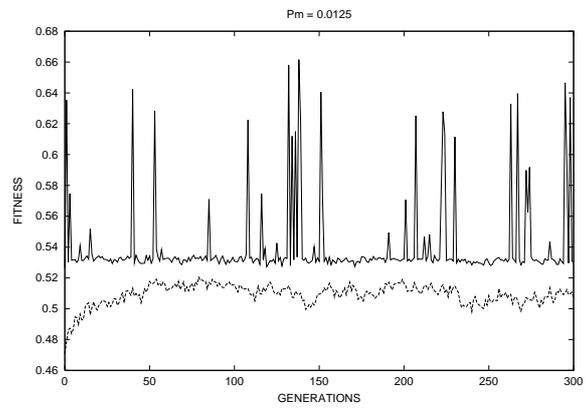
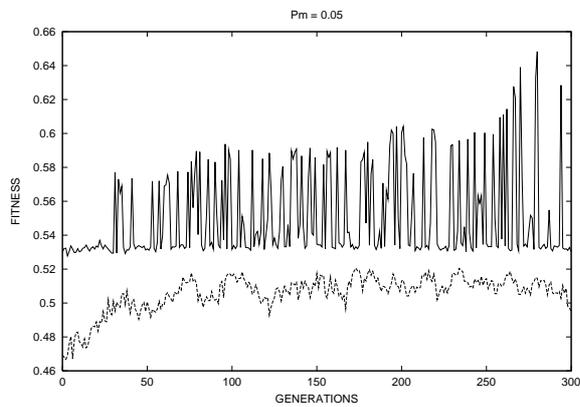
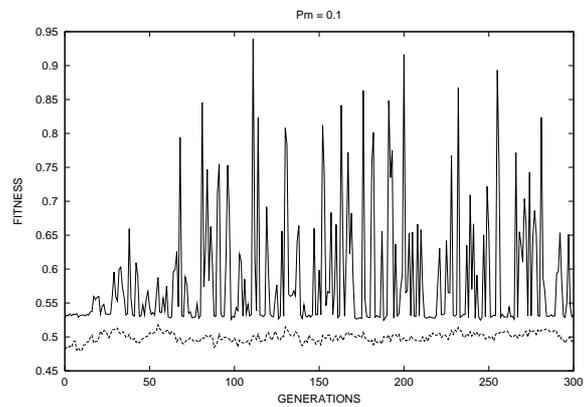
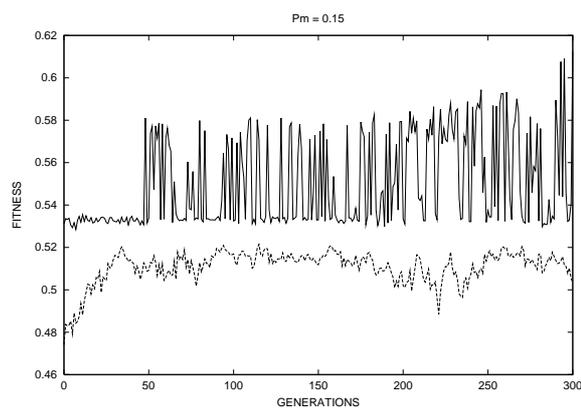
(a) $p_m = 0.005$ (b) $p_m = 0.0125$ (c) $p_m = 0.05$ (d) $p_m = 0.1$ (e) $p_m = 0.15$

FIG. 2.15: Influence du paramètre de mutation p_m sur l'évolution (— Meilleur - - - Moyenne)

vers la modification sur les articulations du manipulateur. Cette solution n'est qu'une amélioration provisoire et ne permet pas de remédier radicalement au problème initial (la plate-forme mobile s'éloigne de la trajectoire).

- la qualité des solutions dès la phase d'initialisation. Lors de la phase d'accouplement, des gènes sont échangés entre les chromosomes. Si la solution n'existe pas dans les solutions proposées dès l'initialisation (un des chromosomes possède une consigne sur la roue gauche du véhicule égale à celle de la roue droite d'un autre chromosome de manière à ce que leur accouplement puisse donner deux consignes égales sur ces deux roues), la réussite de l'optimisation ne repose plus que sur la mutation des gènes (figure 2.15). Celle-ci permet l'introduction de gènes inédits, et par le biais de la reproduction, donne beaucoup plus de chances de trouver une bonne solution.

Il faut donc déterminer la part de mutation à fournir de manière à orienter les recherches de l'algorithme vers l'amélioration de la valeur d'adaptation. L'obtention d'une solution acceptable ($F = 1$ avec la marge d'erreur liée à la discrétisation des variables, F_ε) se doit d'être systématique pour que l'approche proposée soit validée. D'un autre côté, nous avons vu que l'augmentation de ce paramètre provoquait un caractère cahotique dans les résultats, lié à la perte des individus performants d'une génération à l'autre. Quoi qu'il en soit, la valeur initialement définie par la relation

$$p_m = \frac{1}{2 * n_b * \ell}$$

ne permet pas d'exploiter les capacités évolutives à leur maximum.

2.5.2 Perte des bonnes solutions

L'émergence d'une bonne solution ne conduit à rien si celle-ci s'égaré dans les méandres de la mutation et de la reproduction des individus. Sur la figure 2.14 par exemple, on observe quasiment une réponse binaire en fonction des générations (très bonne *ou* très mauvaise solution) tandis que la moyenne de la population suit un contour beaucoup plus régulier (toujours en croissance après 300 générations). Cependant, la solution finale est malheureusement une des mauvaises solutions. La conservation des individus valables au cours des générations doit donc être envisagée de manière à ne pas subir la perte des bonnes solutions.

2.5.3 Caractéristiques de l'évolution

La rapidité de convergence de l'algorithme ne doit pas être négligée. De manière à exploiter totalement notre approche, cette convergence doit avoir lieu dans une marge maximale de 200 générations et de manière systématique. Cette considération est raisonnable puisque, lorsqu'elle apparaît, une solution valable émerge actuellement avant la 230^{ème} génération. Les critères en eux-même doivent être modifiés pour augmenter leur impact sur la recherche. La définition du critère d'erreur en position par exemple

n'est utile que lorsque l'effecteur s'est acquitté de la distance à parcourir. Dans les autres cas, celui-ci revêt un caractère binaire totalement inutile.

2.5.4 Conclusion

Ce chapitre a permis de mettre en évidence des difficultés classiques lors de l'optimisation génétique, sur un exemple simple de synthèse de consignes. La structure robotique utilisée était fixée dans les limites de ce chapitre. Les aspects délicats que nous avons rencontrés peuvent être listés de la manière suivante :

- Interprétation et pose du problème d'optimisation
- Choix du codage et précision sur les solutions
- Adaptation des paramètres génétiques

Ils sont tous liés, directement ou non, à la pose du problème génétique et nécessite une bonne compréhension de la technologie pour être abordés. Le chapitre suivant prend en compte toutes les remarques effectuées pour améliorer l'algorithme et valider les solutions qu'il propose en fin d'évolution. Une fois cette trajectoire simple effectuée systématiquement par un jeu de consignes, le cas où une manœuvre est nécessaire (donc où plusieurs consignes doivent être déterminées) peut être envisagé.

Chapitre 3

Générateur de consignes évolutionnaire

Ce chapitre est destiné à la mise en place d'un générateur d'entrées de commande fiable pour le manipulateur mobile précédemment décrit. Les caractéristiques dynamiques des structures utilisées sont donc toujours fixées. Le suivi de trajectoire sans manœuvre est dans un premier temps résolu. Il s'appuie sur les résultats du chapitre précédent pour générer de manière *systématique, fiable et rapide* cinq consignes qui permettent à l'effecteur de suivre la trajectoire simple précédemment décrite, sans collisions avec l'environnement. La fiabilité et la robustesse de l'algorithme évolutionnaire étant assurée pour cette tâche simple, le cas où le robot doit manœuvrer pour effectuer un suivi est ensuite abordé puis validé, en utilisant trois boucles d'optimisation non indépendantes. Les performances de l'algorithme général sont alors améliorées en supprimant l'une de ces boucles grâce à la manipulation de chromosomes mixtes entiers/réels de longueurs variables.

3.1 Amélioration des performances

Nous nous basons sur l'analyse du chapitre précédent pour remédier aux problèmes rencontrés :

- Interprétation du problème à résoudre
- Choix du codage
- Adaptation des paramètres

Les modifications apportées, décrites ci-après, concernent les points suivants :

- L'augmentation de la précision des solutions en utilisant un codage réel et non binaire
- L'amélioration de l'évaluation des solutions
- L'adaptation des manipulations génétiques, opérateurs et paramètres.

3.1.1 Adaptation du génotype

Le codage binaire utilisé ci-avant a montré ses limites face au problème d'optimisation évolutionnaire. Afin d'obtenir une précision maximale pour les solutions, un codage réel est utilisé pour la construction du génotype. Dans notre cas d'étude, le chromosome est alors constitué de 5 gènes représentés par des nombres réels compris entre 0 et 1. Les modifications conséquentes des opérateurs sont extraites de la littérature [Koza92] [Goldberg89] :

- La sélection reste inchangée, elle est toujours basée sur la valeur de la fonction objective des individus (la nature du codage ne l'influence pas).
- L'opérateur de croisement utilise un croisement uniforme macroscopique. Les valeurs de deux gènes, entre deux parents, sont échangées avec la probabilité de croisement p_c .
- L'opérateur de mutation gère la modification de paramètres réels, en remplaçant la valeur d'un gène par son complémentaire, avec la probabilité de mutation p_m .

Ces différents opérateurs sont plus précisément décrits en annexe 4. L'opérateur de croisement utilisé permet un plus grand brassage des génotypes que celui à deux points de césure du chapitre précédent, et permet donc une plus grande recherche sur l'espace des solutions.

L'utilisation d'un codage réel autorise une plus grande précision sur les variables. Une conséquence est l'augmentation de la taille de l'espace de recherche des solutions (il devient infini). Seule une partie de cet espace va être étudiée, ce qui est le principe de base des algorithmes évolutionnaires. L'équilibre exploration / exploitation de la recherche s'en trouvera de même affectée. Celui-ci est déterminé par les paramètres associés aux différents opérateurs. Il n'existe pas réellement de méthode pour les déterminer, ils sont fixés en fonction de la convergence de l'algorithme. Leurs nouvelles valeurs seront discutées par la suite.

Les conditions d'arrêt de l'algorithme ne prennent plus en compte, du fait du codage, l'erreur de discrétisation précédemment évoquée. Les conditions d'arrêt, pour l'évolution, deviennent :

- Détermination d'une solution dont la valeur d'adaptation est maximale ($F = 1$)
- Atteinte du nombre maximal de génération g

Les conditions d'arrêt pour le test d'un vecteur de consignes restent inchangées. Nous les rappelons :

- Arrivée de l'organe terminal du manipulateur mobile à la hauteur de la position finale désirée ($C_1 = 0$)
- Atteinte de l'erreur maximale autorisée ($\varepsilon = \varepsilon_\ell$)
- Temps de simulation maximal atteint

3.1.2 Fonction d'adaptation

Les modifications apportées à la fonction d'adaptation vont permettre une plus grande efficacité dans l'évaluation des solutions, en évitant la situation présentée sur la figure 2.13 tout en rendant plus actif le critère d'erreur en position au cours de la recherche (figure 2.11). Le premier critère, C_1 , est inchangé. Il représente toujours le pourcentage de trajectoire restant à parcourir à la fin de la simulation :

$$C_1 = \frac{\|X_d - X_f\|}{\|X_d - X_i\|}$$

- X_d : position finale désirée de l'effecteur
- X_i : position initiale de l'effecteur
- X_f : Position atteinte par l'effecteur en fin de simulation

Le critère d'erreur en position C_2 précédemment utilisé, est modifié de manière à avantager les solutions qui, pour une même erreur finale, sont parvenues à un point plus avancé dans la direction de la trajectoire :

$$C_2' = \frac{1}{n} \frac{\sum_{t=0}^{t_f} \varepsilon(t) \times \|X_d - X(t)\|}{\varepsilon_\ell \times \|X_d - X_i\|}$$

- $\varepsilon(t)$: écart entre la position de l'effecteur et la trajectoire désirée à l'instant t
- $X(t)$: position de l'effecteur sur la trajectoire désirée à l'instant t
- X_d : position finale désirée de l'effecteur
- X_i : position initiale de l'effecteur
- n : nombre de pas de temps

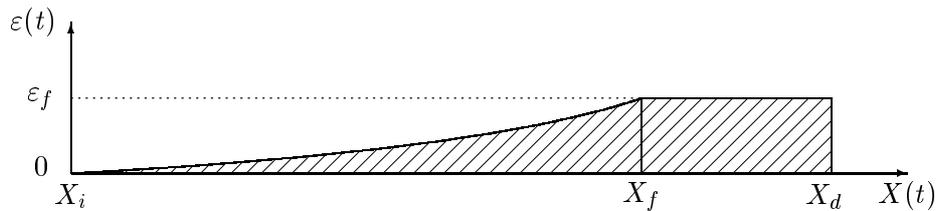


FIG. 3.1: Exemple de surface d'écart à la trajectoire désirée à la fin d'une simulation

Ce critère représente l'aire discrétisée entre les trajectoires réelle et désirée de l'effecteur. Un exemple l'illustre sur la figure 3.1, en fin de simulation d'un vecteur de consignes. La surface est délimitée, dans le plan (ε, X) , par les droites $\varepsilon = 0$ et $\varepsilon = \varepsilon_f$, la courbe d'évolution de l'erreur en position de l'organe terminal $\varepsilon = f(X)$ et la droite $X = X_d$. Outre son caractère simple et l'accessibilité directe aux variables impliquées, cette expression différencie sans ambiguïté la qualité de deux trajectoires ayant la même position finale. La surface $\varepsilon_\ell * (X_d - X_i)$ prise en référence pour l'adimensionnement représente l'aire maximale d'évolution possible de l'effecteur. Ce critère est à minimiser, sa valeur minimale (zéro) étant atteinte pour $\varepsilon(t) = 0 \forall t$, ou $X(t) = X_d \forall t$. Cette

seconde condition n'est jamais vérifiée de par les conditions initiales de la simulation ($X(0) = X_i \neq X_d$). La première condition signifie que l'organe terminal du manipulateur mobile suit la trajectoire désirée tout au long du déplacement. La valeur de ce critère peut dépasser 1, dans le cas d'un départ en sens inverse de la trajectoire désirée. Ceci n'est pas gênant du point de vue de l'évaluation puisqu'une telle solution, si elle se présente, sera immédiatement éliminée de la sélection. L'atteinte de X_d est une des conditions d'arrêt de l'algorithme. Le cas $X(t) > X_d$ n'est donc pas envisagé.

Le critère d'évitement d'obstacle C_3 reste inchangé. La fonction d'évaluation résultante est la même que celle précédemment utilisée, en remplaçant l'expression du critère d'erreur en position C_2 par la nouvelle, C'_2 :

$$F = e^{-\frac{k_1 C_1 + k_2 C'_2 + k_3 C_3}{k_1 + k_2 + k_3}}$$

où k_1 , k_2 et k_3 sont les poids associés à chaque critère, et toujours fixés à 1 pour la simulation. Ainsi, si aucun critère n'est favorisé devant les autres du point de vue des poids k_i (les trois critères sont considérés comme équivalents du point de vue de la réalisation de la tâche), il est à remarquer que la distance parcourue intervient dans les deux critères C_1 et C'_2 . Les solutions d'entrées de commande qui permettent à l'effecteur d'augmenter l'avancement sur la trajectoire, et donc de se rapprocher de l'objectif final X_d , sont donc intuitivement avantagées.

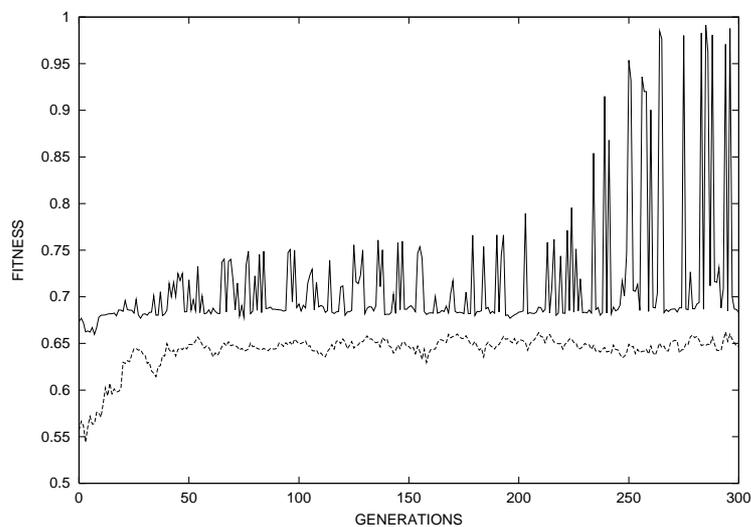


FIG. 3.2: Evolution de la valeur d'adaptation des individus en fonction des générations après modification du critère d'erreur (— Meilleur - - - Moyenne)

Les graphes d'évolution correspondant à ces modifications sont présentés sur la figure 3.2, pour le meilleur individu et pour la moyenne d'une population, sur 300 générations. On retrouve la séparation de la courbe des meilleurs individus que nous avons obtenue en faisant varier la valeur du paramètre de mutation au chapitre précédent

(figure 2.15). En effet, un palier apparaît à $F = 0.68$ tandis qu'une deuxième courbe se dessine croissante jusqu'à des valeurs supérieures à $F = 0.98$ à partir de la génération 260. La perte d'information d'une génération à l'autre est caractérisée dans

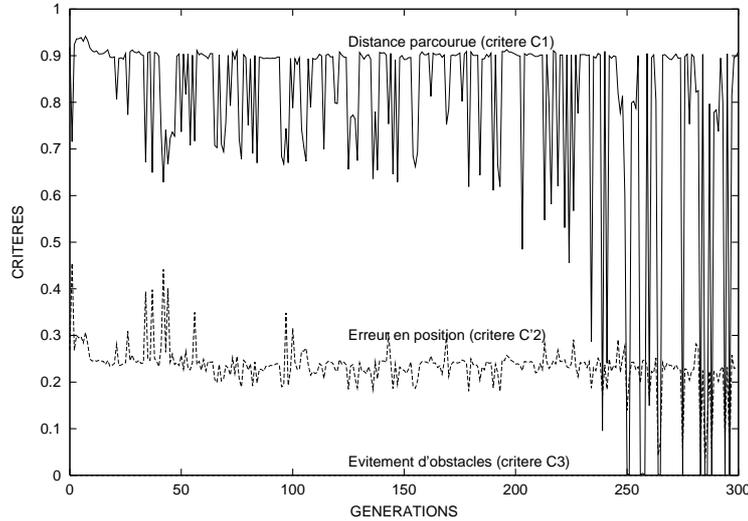


FIG. 3.3: Evolution de la valeur des critères pour les meilleurs individus en fonction des générations après modification du critère d'erreur

cette séparation, dont l'origine sont les très grandes différences de comportement que peuvent apporter de légères modifications sur la valeur d'une entrée de commande (redondance géométrique du système robotique). Toutefois, il est évident que le seul changement du critère d'erreur permet à l'algorithme de converger vers des solutions adaptées à la tâche demandée. Les solutions ambiguës obtenues précédemment (figure 2.13) n'apparaissent plus qu'éventuellement en tout début de simulation mais sont vite définitivement éliminées par la sélection.

L'évolution de la valeur de chaque critère pour le meilleur individu d'une génération nous donne d'autres informations sur le comportement de l'algorithme (figure 3.3). Le critère d'erreur se stabilise à la valeur 0.2 pendant que la distance parcourue augmente très lentement de 1.5 m ($C_1 = 0.7$) à 2 m ($C_1 = 0.6$). Celle-ci augmente réellement à partir de la génération 200 et l'effecteur atteint sa cible la première fois à la génération 250. Le critère d'erreur reprend sa croissance vers la génération 225 pour permettre d'atteindre la cible avec 3.1 cm d'erreur à la génération 264. Il y a donc complémentarité des deux critères C_1 et C_2 dans le sens de l'optimisation. Leur dépendance à notion commune (l'avancement dans la trajectoire) les mène tous deux vers le résultat escompté (réduire la distance restant à parcourir). Cette redondance du point de vue de l'évaluation permet d'accentuer la recherche sur l'objectif tout en orientant la qualité de sa réalisation.

Ces résultats sont très encourageants mais encore insuffisants. Il existe toujours des cas où la solution recherchée n'émerge pas. La proportion de telle situation est descen-

due toutefois à 10 % des évolutions (contre 18 % précédemment, et toujours sur 100 simulations). Deux autres points s'avèrent gênants : d'une part, il faut attendre 250 générations pour obtenir le jeu d'entrées de commande recherché. La solution émerge généralement entre les générations 180 et 250, ce qui est trop long. D'autre part, la perte d'information d'une génération à l'autre a des conséquences désastreuses. Dans l'exemple précédent (figures 3.2 et 3.3), la solution proposée en fin de simulation se voit allouer la valeur d'adaptation $F = 0.68$. Ce résultat final ne tient absolument pas compte du fait que la cible a été atteinte au cours de l'évolution. Il nous faut donc à présent résoudre ces trois problèmes : émergence systématique d'une solution recherchée, conservation des meilleurs individus d'une génération à l'autre et accélération de la convergence.

3.1.3 Mémoire évolutionnaire

La sauvegarde des caractéristiques du meilleur individu d'une évolution, toutes générations confondues, est pratique courante lors de l'optimisation évolutionnaire de manière à éviter la perte de ses caractéristiques d'une génération à l'autre. Cette méthode permet de parer la perte définitive de la meilleure solution après manipulations par les opérateurs évolutionnaires, et les faire participer à la reproduction aux générations suivantes (exploration autour de cette solution). L'utilisation de cette sauvegarde est effectuée ici de manière particulière : la mise en place d'un module de mémorisation des individus les mieux adaptés est effectuée. Il semble cependant que se limiter au seul meilleur individu n'est pas optimal du point de vue de l'évolution, comme le montre les résultats suivants. Ce module consiste donc à récupérer les caractéristiques des solutions présentant les meilleures valeurs d'adaptation de toutes les générations depuis le début de l'évolution et forcer leur ré-introduction dans les générations suivantes. A chaque génération, les nouvelles meilleures solutions sont comparées à celles déjà enregistrées, issues des générations précédentes. La liste est éventuellement modifiée de manière à ce que la quantité de meilleures solutions mémorisées reste constante. L'ensemble de ces solutions plus performantes est automatiquement réintroduit dans la sélection pour la génération suivante. Si la solution considérée y est déjà présente, le reste de la population est inchangé. Dans le cas contraire, le nouvel individu remplace le moins performant de la génération. La taille de la population reste ainsi constante.

Le module utilisé entraîne l'apparition d'un nouveau paramètre évolutionnaire, p_r , lié au nombre des meilleurs individus dont les caractéristiques seront conservées au cours des générations. Celui-ci ne peut être qu'un entier dont la valeur minimale (0) correspond aux cas précédents (simulation inchangée), et la valeur maximale (k , taille de la population) présente une invariance de la population au cours des générations (toute la population est ré-introduite dans la génération suivante). La figure 3.4 montre un exemple de simulation aboutissant à une solution d'adaptation maximale, dans le cas où les trois chromosomes présentant la meilleure valeur d'adaptation depuis le début de la simulation sont ré-introduit d'une génération à la suivante. Ces graphes d'évolution montrent une progression par paliers successifs de la valeur d'adaptation du meilleur

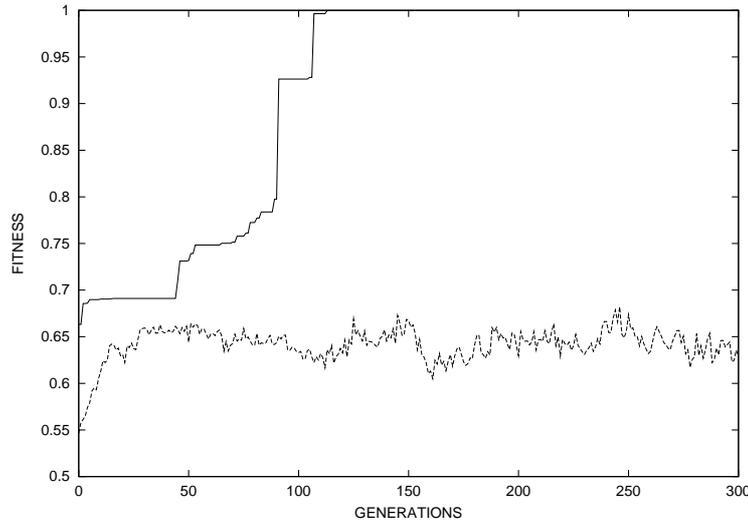
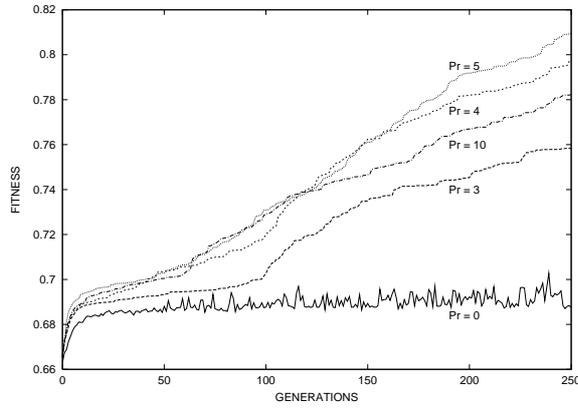


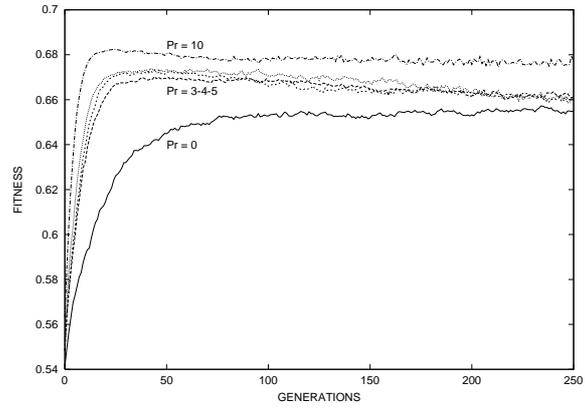
FIG. 3.4: Graphes d'évolution utilisant l'opérateur de ré-introduction des meilleurs individus (— Meilleur - - - Moyenne)

individu, partant de la valeur classique de $F = 0.66$ à sa valeur maximale $F = 1$. La condition d'arrêt de l'algorithme n'a pas été utilisée ici de manière à observer le comportement de l'adaptation moyenne de la population. La figure 3.4 met en évidence la croissance rapide de cette dernière courbe pendant les 20 premières générations, de 0,55 à 0,67. Elle présente ensuite un palier stable avant de suivre une décroissance très lente, avec stabilisation autour de la valeur 0,64. Ce comportement va à l'encontre des allures classiques d'évolutions, lesquelles montrent une croissance très lente jusqu'à stabilisation autour de la valeur de convergence. Les statistiques effectuées sur les évolutions, présentées par la suite, confirment cette observation.

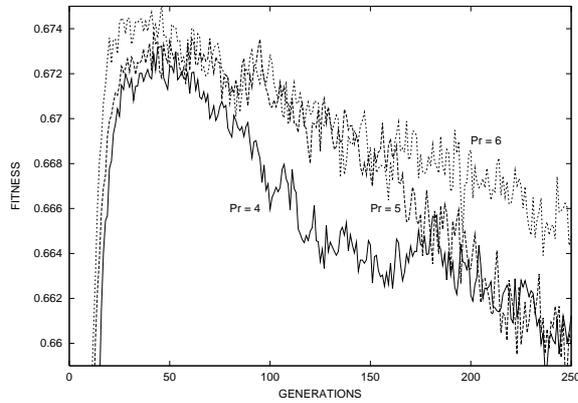
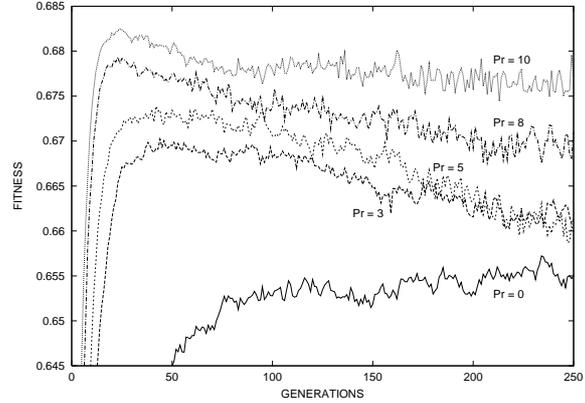
L'influence du paramètre de récupération p_r sur l'évolution est présentée sur la figure 3.5, pour des valeurs échantillons comprises entre $p_r = 0$ (cas sans récupération) et $p_r = 10$ (i.e. les 10 meilleurs sont ré-introduits à la génération suivante). Ces courbes concernent l'évolution de la fitness des meilleurs individus au cours de 250 générations. La valeur moyenne inférieure à 1 de la fonction d'adaptation provient des évolutions non abouties. La première remarque provoquée par ces résultats porte sur le fait que l'adaptation du meilleur individu (figure 3.5(a)) est statistiquement toujours en croissance régulière à la 250^{ème} génération (ces statistiques concernent 100 simulations différentes). On retrouve, sur l'allure des courbes de la figure 3.5(a), le départ classique d'évolution, jusqu'à la quarantième génération environ, évoqué dans le paragraphe précédent. Les valeurs de la fonction d'adaptation varie en fonction du paramètre p_r . Au lieu de ce stabiliser, cette valeur reprend alors sa croissance d'une manière régulière, selon une pente qui dépend du paramètre p_r . L'évolution de l'adaptation moyenne des populations (figure 3.5(b)) subit un schéma de convergence inverse, avec une pente de décroissance d'autant plus importante que la croissance du meilleur individu est forte. Les figures 3.5(c) et 3.5(d) sont des agrandissements des évolutions de la fit-



(a) Meilleurs individus

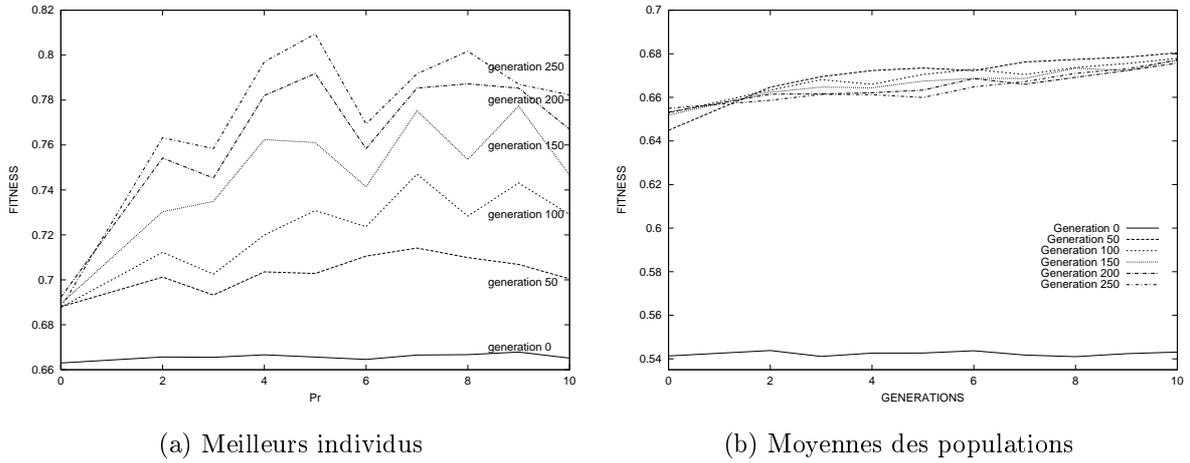


(b) Moyennes des populations

(c) Moyennes des populations (agrandissement), $P_r = 4$, $P_r = 5$ et $P_r = 6$ (d) Moyennes des populations (agrandissement), $P_r = 0$, $P_r = 3$, $P_r = 5$, $P_r = 8$ et $P_r = 10$ FIG. 3.5: Influence du paramètre de mémorisation sur l'évolution avec $p_m = 0.0125$

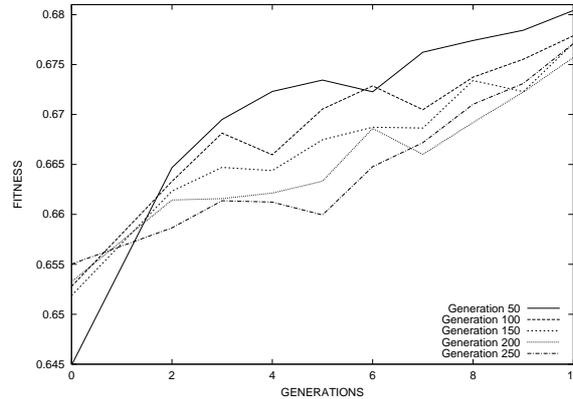
ness moyenne des populations pour différentes valeurs de p_r et illustrent les remarques précédentes. L'algorithme montre clairement de meilleurs résultats pour le meilleur individu pour la valeur $p_r = 5$. L'évolution de l'adaptation du meilleur individu, dans ce cas, est de croissance supérieure aux autres cas (figure 3.5(a)). En revanche, L'adaptation de la moyenne des individus de la population possède la plus grande pente de décroissance (figure 3.5(b)). Le choix de la valeur du paramètre de récupération, est fixé par la valeur maximale de la fonction $F(p_r)$ présentée sur la figure 3.6(a). Celle-ci est issue des courbes d'adaptation précédentes. Elle montre un schéma de croissance similaire au cours des générations, depuis le départ des simulations où toutes les valeurs d'adaptations sont identiques quel que soit p_r . Elle présente deux optima, pour $p_r = 5$ et $p_r = 8$, ce dernier révélant des résultats performant à partir de la génération 200. La vitesse de convergence étant un caractère essentiel de notre approche, le paramètre

est fixé à 5 pour tout le reste de l'étude. La section suivante discutera des interactions qui lient les différents paramètres évolutionnaires entre eux.



(a) Meilleurs individus

(b) Moyennes des populations



(c) Moyennes des populations (agrandissement)

FIG. 3.6: Statistiques de la valeur d'adaptation en fonction de p_r et de la génération

L'introduction de ce module de récupération des meilleurs individus a donc deux conséquences positives sur l'évolution. La première, qui était l'objectif, est la conservation des caractéristiques, d'une génération à la suivante, d'une solution qui parvient à avancer dans la réalisation de la tâche mieux que les autres. La deuxième, qui en découle directement, montre une accélération dans la convergence de l'algorithme vers une solution adaptée au problème. En effet, il fallait attendre précédemment 250 générations pour être sûr d'avoir une solution. Cette attente se limite à 187 en utilisant la conservation des caractéristiques (sur 100 simulations effectuées).

Ce module est considéré comme un nouvel opérateur, de mémorisation, avec son paramètre de récupération des individus p_r . Ce paramètre n'est pas une probabilité de ré-introduction mais une constante de l'algorithme. Il subit cependant une influence réciproque avec les autres paramètres évolutionnaires (taille de la population, probabilités de croisement et de mutation, nombre de générations), comme nous allons le voir.

3.1.4 Dynamique et mémoire évolutionnaires

L'opérateur de mémoire évolutionnaire précédemment introduit peut être associé à un phénomène culturel au sein d'une société, la conservation d'une mémoire des individus cités en exemple. Le fait de retrouver d'une génération à l'autre les caractéristiques de différents individus mieux adaptés que les autres autorise une augmentation conséquente de la dynamique évolutionnaire, directement liée aux probabilités de croisement et de mutation. On a vu, toutefois, qu'un compromis est à déterminer entre la dynamique et la convergence évolutionnaire. Un avantage supplémentaire de l'"objectif culturel" est qu'il permet, comme nous allons le voir, de considérablement augmenter la dynamique évolutionnaire tout en conservant à l'évolution un caractère stable.

L'effet du paramètre de mutation p_m est de permettre l'émergence de nouvelles solutions, quelle que soit leur valeur d'adaptation. Dans le cadre de notre étude précédente

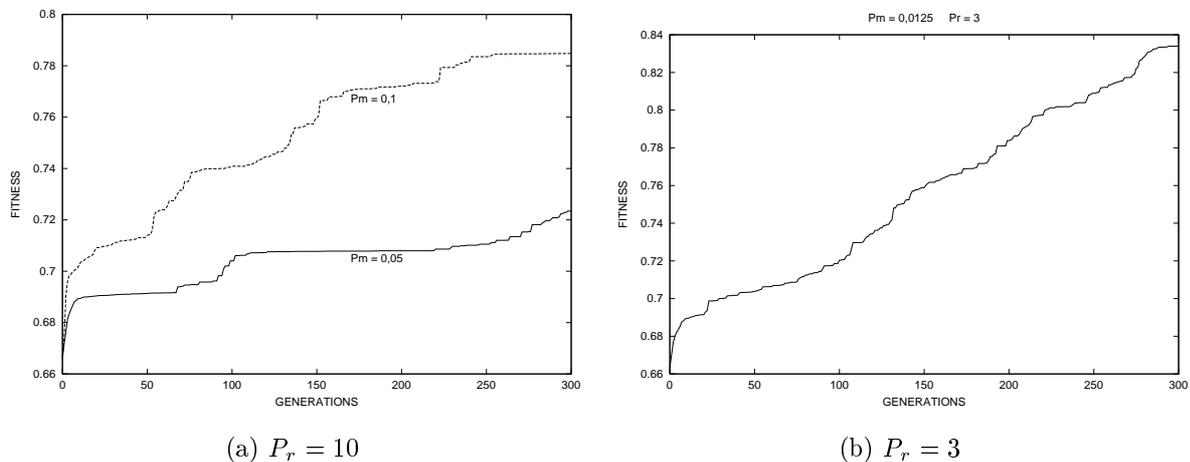


FIG. 3.7: Influence du paramètre de mutation sur l'évolution avec $p_r = 10$ et $p_r = 3$ pour les meilleurs individus

(Fig. 2.15), nous avons constaté l'émergence plus fréquente de bonnes solutions dans le cas $p_m = 0,1$. Cette valeur est retrouvée ici, comme le montrent les deux graphes de la figure 3.7. Ces courbes reproduisent l'évolution du meilleur individu au cours des générations et en fonction du paramètre de mutation pour des valeurs variant entre

0,0125 et 0,15. Elles mettent en évidence l'indépendance entre p_m et p_r puisque les mêmes valeurs optimales sont retrouvées pour deux valeurs différentes de p_r .

Le paramètre de croisement, quant à lui, peut être mis à son maximum puisque la conservation aléatoire d'individus ne se pose plus. La totalité des individus d'une

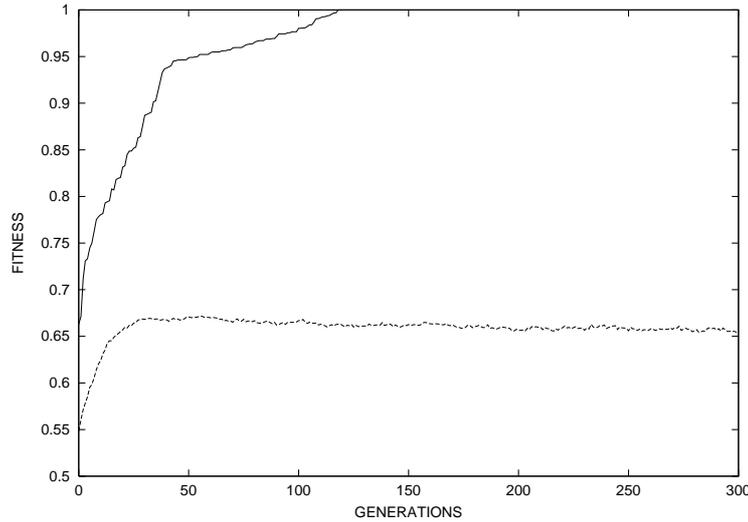


FIG. 3.8: Courbes d'évolution finales (— Meilleur - - - Moyenne)

population est donc modifiée et aucun ne sort indemne des manipulations évolutives excepté les p_r meilleurs toutes générations confondues. La courbe d'évolution présentée sur la figure 3.8 correspond à l'évolution finale moyennée sur 100 évolutions pour les meilleurs individus et pour la moyenne de la population. Les paramètres, pour ces évolutions, ont été fixés à leur valeurs optimales du point de vue de la rapidité de convergence (tableau 3.1). Cette courbe présente un point de flexion au niveau de la

Paramètre	Valeur
k	100
g	300
p_m	0.11
p_c	1
p_r	5

TAB. 3.1: Paramètres optimaux pour la convergence à une valeur d'adaptation maximale

génération 45. A ce moment, 82 évolutions sur les 100 ont atteint la valeur maximale de la fonction d'adaptation. Les 18 % restant se répartissent jusqu'à la 117^{me} générations où cette valeur est atteinte pour toutes les évolutions. La valeur moyenne de la population se stabilise très rapidement (30^{ème} génération, résultant du remaniement complet de la population à chaque génération (figure 3.1). On peut dire à ce niveau

que l'objectif de l'introduction de cet opérateur a été rempli. Son inconvénient majeur réside dans l'introduction d'un nouveau paramètre à déterminer. Ses avantages sont l'augmentation de la vitesse de convergence et surtout l'aboutissement systématique de la recherche.

3.1.5 Conclusion

La première mise en place de l'algorithme évolutionnaire, effectuée "dans les règles", s'est avérée peu adaptée à la tâche désirée (chapitre 2). Convergence prématurée voire absence totale de convergence, perte des bonnes solutions d'une génération à l'autre sont apparues comme autant de caractéristiques de ces évolutions initiales. L'étude réalisée au chapitre précédent a permis de mettre en évidence le rôle non négligeable joué par la redondance géométrique de la structure mécanique utilisée, donc, par extension, entre les solutions manipulées par l'algorithme (les entrées de commande). La qualité de l'évaluation, désirée la plus simple dans l'expression des critères et dans leur nombre de manière à garder l'objectivité de la sélection, s'est aussi montrée mal adaptée aux besoins de l'algorithme.

Les modifications apportées dans ce début de chapitre ont permis de remédier à ces faiblesses. Celles-ci concernent

- La prise en compte de l'historique de la trajectoire et plus seulement de la position finale atteinte ;
- L'introduction d'un nouvel opérateur évolutionnaire et de son paramètre ;
- L'adaptation au problème de la valeur des paramètres évolutionnaires.
- L'utilisation directe de valeurs réelles pour les solutions manipulées par l'algorithme.

Ces changements ont apporté les améliorations nécessaires pour assurer la fiabilité de l'évolution face au problème posé (suivi d'une trajectoire sans collision avec une entrée de commande constante sur chaque actionneur). En effet les moyennages effectués montrent l'émergence d'une solution valable, systématiquement et dans un délai maximal de 120 générations. Les effets des modifications sont essentiellement les suivants :

- La disparition, dès le début de l'évolution, des solutions ambiguës du point de vue de l'évaluation provoquant des trajectoires erronées et conservées dans la sélection. Cette élimination a permis de limiter les manipulations évolutionnaires uniquement aux solutions fiables ;
- La conservation des bonnes solutions et leur ré-introduction forcée dans la sélection d'une génération à l'autre, de manière à assurer la cohérence de la généalogie de l'évolution. L'apparition d'une bonne solution devient systématique et plus rapide.
- L'adaptation au problème de la valeur des paramètres évolutionnaires augmente encore la rapidité de convergence de l'algorithme.
- L'utilisation directe de valeurs réelles pour les solutions augmente la précision des consignes (suppression de l'erreur de discrétisation).

Ces résultats très positifs nous permettent d'envisager la recherche des vecteurs d'entrées de commande dans le cas d'une trajectoire nécessitant une manœuvre. Celle-ci est alors considérée comme résultant de la succession de vecteurs d'entrées de commande constants par morceaux.

3.2 Cas de la manœuvre

3.2.1 Position du problème

La détermination d'une commande optimale du point de vue de la réalisation de la tâche fixée nécessite l'intervention de nouveaux paramètres afin d'adapter le problème précédent au cas incluant une manœuvre. Demander à l'algorithme de chercher les consignes constantes, qui mènent le robot le plus loin possible sur la trajectoire, par exemple, peut s'avérer problématique sur l'ensemble de la trajectoire. En effet, nous ne

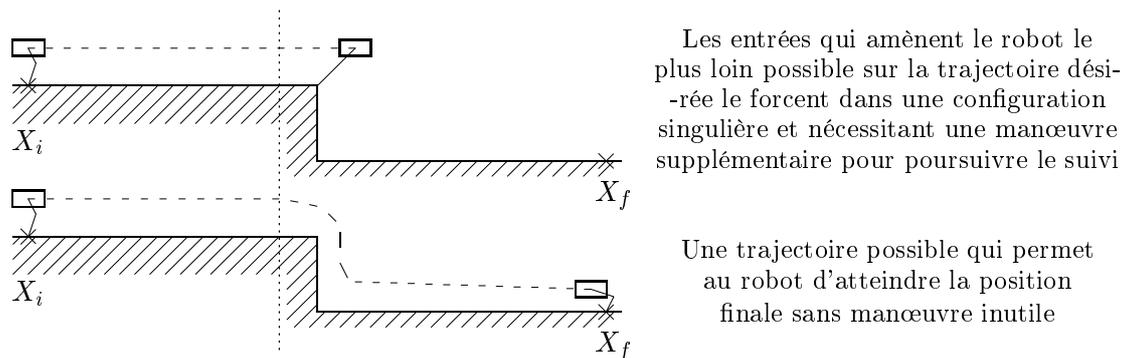


FIG. 3.9: Recherche d'une trajectoire "optimale" dans le cas d'un suivi de paroi

pouvons pas affirmer qu'un ensemble d'entrées de commande est optimal du point de vue de l'ensemble de la tâche parce que celui-ci mène le robot plus loin que les autres sur une partie de la trajectoire. La figure 3.9 montre un exemple qui peut être tout à fait courant : le schéma supérieur considère le cas évoqué ci-avant. L'effecteur du robot atteint une situation très avancée sur la trajectoire désirée. Malheureusement, le manipulateur mobile se retrouve dans une configuration qui le force à une manœuvre pour pouvoir continuer le suivi. Cette manœuvre diminue ses performances vis-à-vis de la tâche dans son ensemble. Le schéma inférieur montre une solution qui permet d'atteindre la position finale avec une trajectoire continue. La trajectoire précédente est interrompue avant de le placer dans une configuration difficile à gérer pour continuer. Cette interruption lui permet d'effectuer un virage pour s'acquitter d'une trajectoire continue. Nous devons donc considérer une évaluation locale qui assure la minimisation instantanée de l'erreur en situation. Une deuxième optimisation doit être mise en place de manière à évaluer globalement la qualité du suivi. La première concerne la réalisation effective de la tâche tandis que la deuxième quantifie la manière dont le robot s'en

acquies. L'hétérogénéité de ces deux types d'évaluation nous forcent à effectuer -au moins- deux boucles d'optimisation différentes.

3.2.2 Paramètres influents

Nous avons évoqué dans le paragraphe précédent deux contraintes permettant de quantifier en partie la valeur d'une commande. La première, locale, vérifie la réalisation de la tâche. La seconde, globale, limite le nombre de manœuvres pour réaliser cette tâche. Elle suppose un minimum de consignes différentes sur la totalité du suivi. Nous pouvons donc considérer une commande comme performante lorsqu'elle permet à l'effecteur de *suivre une trajectoire désirée d'une position initiale à une position finale données, moyennant un minimum d'entrées de commande constantes différentes appliquées aux actionneurs*. Le nombre d'entrées de commande différentes est donc

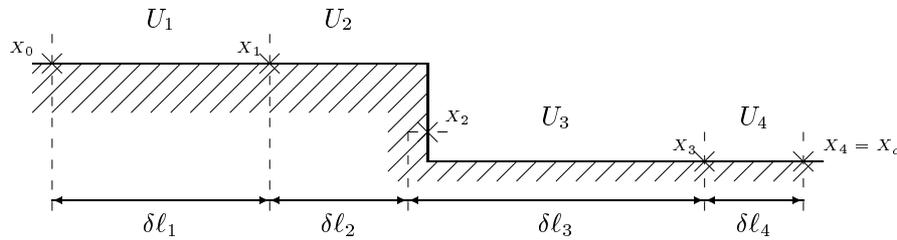


FIG. 3.10: Une répartition temporelle des vecteurs d'entrées de commande dans le cas $n = 4$.

à minimiser. Cette dernière affirmation présente un deuxième avantage : un voltage constant dont la valeur change peu fréquemment diminue les sollicitations sur le moteur auquel elle s'applique. Cela permet de diminuer la consommation et l'usure de celui-ci.

L'approche que nous adoptons est schématisée sur la figure 3.10. Elle est influencée par trois paramètres :

- La quantité de vecteurs d'entrées de commande différents nécessaires pour suivre une trajectoire dans sa totalité ;
- Le temps d'application de chaque vecteur de consignes ;
- Les valeurs des entrées de commande appliquées sur chaque actionneur du robot et qui permettent de suivre la trajectoire désirée.

Les entrées de commande sont à déterminer de manière à ce que l'erreur en situation soit minimale à tout instant. Cette approche soulève une question fondamentale concernant la détermination de l'intervalle de temps pendant lequel un ensemble d'entrées de commande sera appliqué avant de passer à l'ensemble suivant. Ceci implique une répartition des intervalles sur la trajectoire, donc une détermination de la division de la trajectoire en n sous-trajectoires qui peuvent être considérées comme quasi-indépendantes, où n représente le nombre d'entrées de commande différentes. Cette répartition dépend

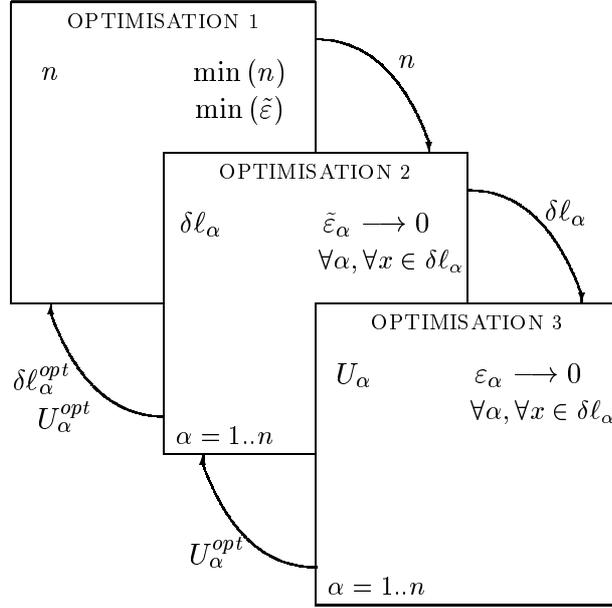


FIG. 3.11: Génération d'une commande par optimisations imbriquées.

de la difficulté locale de réalisation du suivi et de l'adaptation géométrique et cinématique du robot étudié. La figure 3.10 illustre la décomposition géographique d'une trajectoire désirée en quatre subdivisions de longueur $\delta l_{1..4}$. Chaque intervalle correspond à une distance à parcourir par l'application du vecteur d'entrées de commande $U_{1..4} = (u_1, \dots, u_{n_b})^t$ où n_b représente le nombre de degrés de libertés du robot étudié ($n_b = 5$ dans notre cas). Les sous-trajectoires sont liées par leurs points de départ et d'arrivée, X_0, X_1 à X_n , mais chaque vecteur d'entrées de commande sur une subdivision est déterminé indépendamment des autres subdivisions. Chacun des groupes de paramètres va faire l'objet d'une optimisation à part entière (Fig. 3.11). La commande recherchée est donc le résultat de trois optimisations non indépendantes :

- minimisation du nombre d'entrées de commande différentes n .
- La répartition des intervalles géographiques d'application δl_α ($\alpha = 1..n$) des entrées de commande, à déterminer en fonction de la difficulté locale du suivi.
- La minimisation l'erreur en situation de l'effecteur permettant de déterminer l'ensemble de consignes U_α ($\alpha = 1..n$) à appliquer sur les n_b actionneurs du robot.

Ces trois paramètres ne sont pas indépendants et forcent une optimisation sur trois niveaux. Nous pouvons maintenant décrire précisément l'algorithme permettant de déterminer complètement la commande dans le cas du manipulateur mobile à 5 ddl utilisé précédemment.

3.2.3 Optimisations imbriquées

L'algorithme général est présenté à la figure 3.12. L'optimisation des trois groupes de paramètres est présentée comme une optimisation sur trois boucles imbriquées. L'optimisation notée 1 est globale sur l'ensemble de la simulation. Elle tient compte

des résultats des deux boucles inférieures et évalue le nombre n de consignes différentes utilisées pour effectuer le suivi. L'optimisation 2 utilise les données des deux boucles qui l'entourent. La dimension du chromosome représentant les solutions dépend de la valeur de n fournie par l'optimisation 1. Elle s'effectue sur la répartition géographique $\delta\ell_\alpha$ ($\alpha = 1..n$) de la trajectoire. Pour un nombre n d'intervalles généré, une répartition géographique de ces intervalles est proposée. Leur valeur est stockée dans les variables

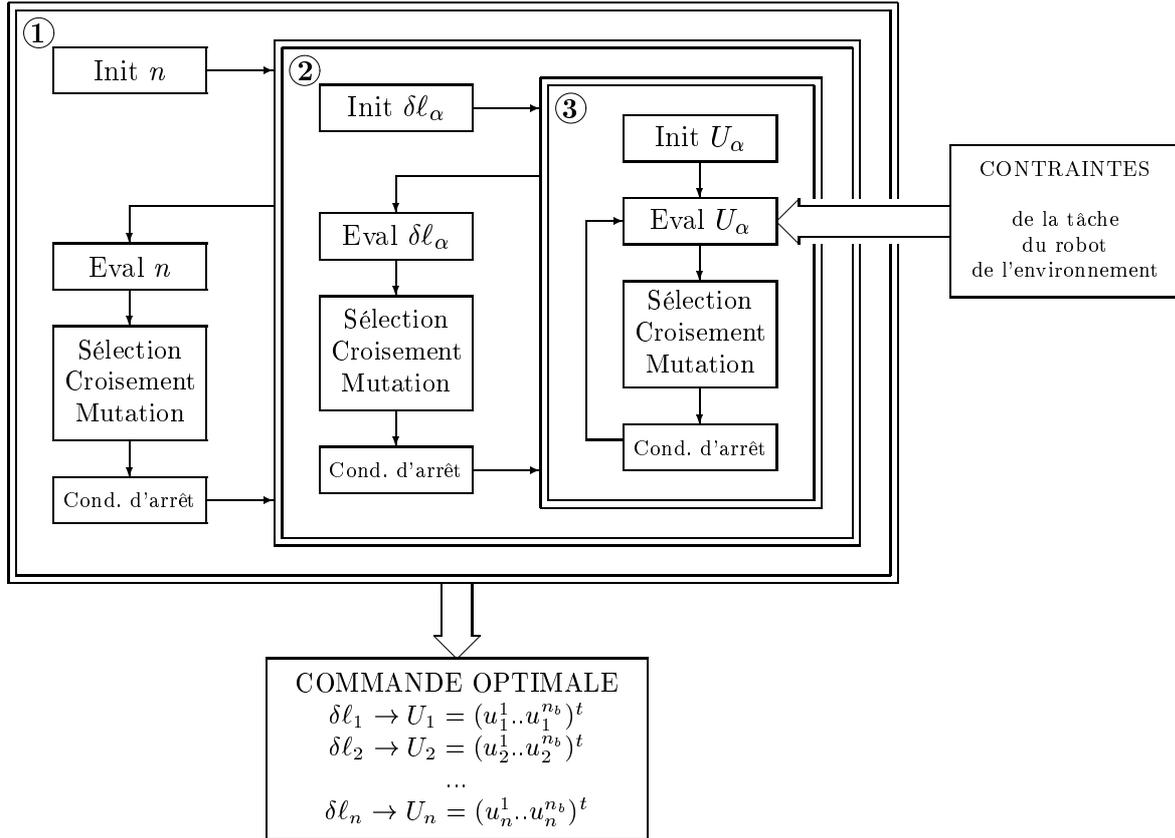


FIG. 3.12: Algorithme général pour la détermination d'une commande évolutionnaire optimale.

$\delta\ell_\alpha$, ($\alpha = 1..n$). Elle ne peut donc commencer que lorsque le nombre d'intervalles divisant la trajectoire est connu. L'optimisation 3 recherche l'ensemble des entrées de commande qui seront appliquées entre les points qui déterminent les extrémités de chacun de ces intervalles géographiques et permettant de suivre la trajectoire à tout instant. A chaque intervalle correspond un ensemble d'entrées réunies dans le vecteur U_α . La deuxième optimisation concerne donc des tronçons de trajectoire tandis que la troisième est une optimisation instantanée. Les trois niveaux d'optimisation dépendent de l'erreur en position pour garantir la stabilité de la commande retenue sur l'ensemble du déplacement. Les contraintes liées au robot et à l'environnement sont prises en compte lors de la recherche du vecteur des entrées du système. Elles interviennent donc directement dans l'évaluation de celles-ci.

3.2.4 Tâche et robots

Les conditions pour la simulation sont globalement les mêmes que précédemment en ce qui concerne les modèles utilisés et la tâche. Le manipulateur mobile est toujours constitué de la plate-forme mobile RoMaiN et du bras manipulateur PUMA 560. La tâche considère, en plus de la précédente, une marche sur la trajectoire comme représenté sur la figure 3.13. Elle consiste à suivre une trajectoire rectiligne horizontale sur 4 mètres puis franchir un renforcement de 50 cm avant une autre droite horizontale de 4 mètres (Fig. 3.13). La configuration initiale du manipulateur mobile est identique

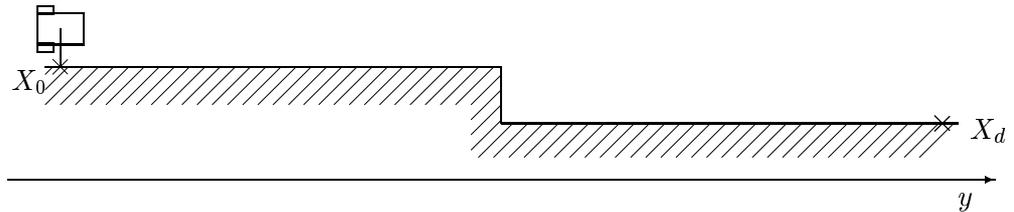


FIG. 3.13: Suivi de trajectoire avec franchissement de marche

à celle du cas précédent. La plate-forme part parallèlement au mur et dirigée dans la direction de la cible. L'effecteur est en $X_0 = (5, 0, 1)$, soit au départ de la trajectoire désirée. Sa position finale doit être en $X_d = (4.5, 8, 1)$.

3.3 Algorithme évolutionnaire

Cette section décrit les différentes étapes pour la constitution de l'algorithme évolutionnaire. Bien que similaires à ceux décrits dans l'étude précédente, les opérateurs évolutionnaires sont rappelés ainsi que les paramètres qui leur sont associés. La définition des variables à déterminer ainsi que l'évaluation des solutions proposées dans les différentes optimisations sont décrites précisément dans un premier temps.

3.3.1 Représentation des solutions

Nous devons définir clairement les trois chromosomes (supérieur, intermédiaire, inférieur, voir figure 3.12) qui seront manipulés par les algorithmes : le nombre d'intervalles divisant la tâche, la longueur de ces intervalles et les entrées de commande appliquées sur les actionneurs du robot pendant chaque intervalle (figure 3.10).

Le chromosome supérieur n représente le nombre d'intervalles qui divisent la trajectoire globale. Il contient un entier codé sur 3 bits, ce qui permet 8 possibilités pour la division. Cette quantité est suffisante pour les trajectoires simples que nous allons tester. Il doit de toutes façons rester de faible valeur dans le sens où n détermine la longueur des chromosomes des étages inférieurs (tableau 3.2).

	Codage	Dimension chromosome	Valeurs max
n	Entier, binaire	3 bits	8
$\delta\ell_i$	Réel $\in [0, 1]$	n	10 m
u_i	Réel $\in [0, 1]$	5	b_i N.m

TAB. 3.2: Caractéristiques du codage

Les subdivisions de la trajectoire sont représentées par des nombres réels. La dimension du chromosome de répartition $\tilde{\delta\ell} = \langle \tilde{\delta\ell}_1.. \tilde{\delta\ell}_n \rangle$, correspondant au vecteur de répartition géographique adimensionné, est donc définie par la valeur du nombre d'intervalles n . Les valeurs représentées par les gènes sont majorées à 10 mètres lors de leur décodage (tableau 3.2). Il est donc possible que la somme des distances soit supérieure à la distance totale à parcourir. Ce problème doit être pris en compte lors de l'évaluation des variables intermédiaires.

Le chromosome inférieur représente le vecteur d'entrées de commande qui sera appliqué pour parcourir la subdivision correspondante. Il est le regroupement de 5 valeurs réelles, où 5 est le ddl de la structure robotique étudiée. Le temps alloué pour la détermination de chaque vecteur d'entrées de commande est toujours de 5 minutes. La solution pour la commande contient donc n vecteurs d'entrées, soit un par intervalle géographique. Outre l'avantage de traiter des chromosomes de tailles inférieures, cette répartition permet de gérer en parallèle l'optimisation de toutes les entrées de commande pour une répartition donnée, soit d'économiser du temps de simulation. L'initialisation des variables intermédiaires (donc la quantité d'optimisations inférieures) et la condition d'arrêt inférieure sont directement dépendantes de la valeur des variables des niveaux supérieurs. Nous allons à présent discuter de l'évaluation des solutions manipulées par les trois algorithmes.

3.3.2 Evaluation des solutions

L'évaluation globale des entrées de commandes proposées par l'algorithme doit tenir compte des contraintes liées

- à la tâche
- à l'environnement
- au robot

Ces trois contraintes sont toutes liées à réalisation ou à la performance dans la réalisation de la tâche. La première concerne sa réalisation brute (l'effecteur suit bien la trajectoire désirée) qui est l'objectif central du robot. La deuxième vérifie sa faisabilité par rapport à l'environnement (évitement des collisions). La troisième évalue le "confort" du robot au cours de ses mouvements (pas de renversement, évitement des singularités, ...). Celle-ci est déjà intégrée dans la définition même de notre problème et est évaluée par l'intermédiaire de la minimisation de la variable globale n . Les deux

premières sont celles explicitées aux chapitres précédents. Nous distinguons les trois évaluations :

- inférieure
- intermédiaire
- supérieure

correspondant à chaque boucle d'optimisation.

Evaluation inférieure : Les trois critères utilisés sont donc sensiblement les mêmes que ceux utilisés précédemment, la méthode différant pour l'évitement d'obstacle (critère C_3). Leur influence sur l'évaluation est schématisée sur la figure 3.14 et leur expression est rappelée ci-après :

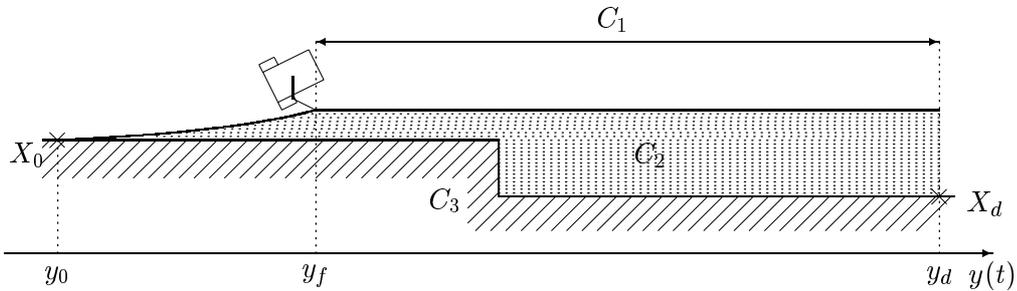


FIG. 3.14: Schématisation des critères pris en compte dans les optimisations inférieure et intermédiaire

- le pourcentage de trajectoire restant à parcourir à la fin de la simulation :

$$C_1 = \frac{\|X_d - X_f\|}{\|X_d - X_i\|}$$

X_d : position finale désirée de l'effecteur

X_i : position initiale de l'effecteur

X_f : Position atteinte par l'effecteur en fin de simulation estimée par le MDD

- L'aire discrétisée entre les trajectoires réelle et désirée de l'effecteur :

$$C_2' = \frac{1}{n_p} \frac{\sum_{t=0}^{t_f} \varepsilon(t) \times \|X_d - X(t)\|}{\varepsilon_\ell \times \|X_d - X_i\|}$$

$\varepsilon(t)$: écart entre la position de l'effecteur et la trajectoire désirée à l'instant t

$X(t)$: position de l'effecteur sur la trajectoire désirée à l'instant t

X_d : position finale désirée de l'effecteur

X_i : position initiale de l'effecteur

n_p : nombre de pas de temps

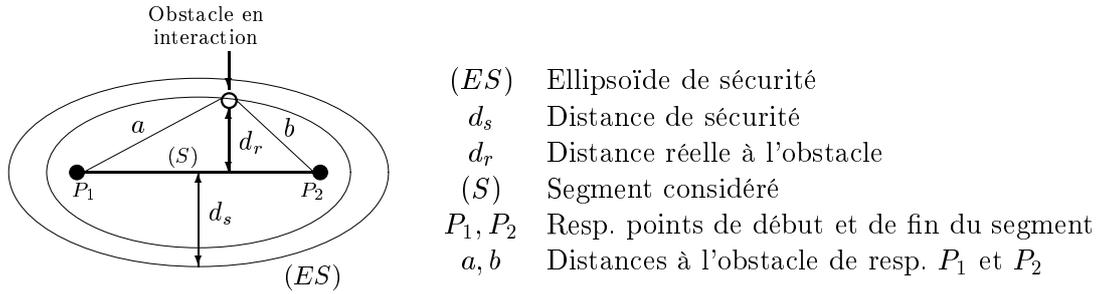


FIG. 3.15: Ellipsoïde d'évitement d'obstacles

- Eloignement des obstacles par la méthode de l'ellipsoïde. Nous ne pouvons plus assurer une non collision uniquement pour les extrémités des segments du robot. Le principe de ce critère est de dessiner une ellipsoïde autour des corps du manipulateur mobile, dite ellipsoïde de sécurité (Fig. 3.15). Le critère d'évitement s'écrit alors :

$$C'_3 = \sum_{j=1}^{n_s} \sum_{i=1}^{n_o} cp_{ji} \quad \text{avec} \quad \begin{cases} \text{Si } Obstacle_i \in ES_j \Rightarrow cp_{ji} = 0 \\ \text{Si } Obstacle_i \notin ES_j \Rightarrow cp_{ji} = \frac{(d_s - d_{ji})}{d_s} \end{cases}$$

n_s : Nombre de segments considérés

n_o : Nombres d'obstacles en interaction

ES_j : Ellipsoïde de Sécurité du segment j

cp_{ji} : Coefficient de coût lié au $j^{\text{ème}}$ segment et pour le $i^{\text{ème}}$ obstacle

d_{ji} : distance réelle du segment j à l'obstacle i

L'interaction d'un obstacle dans l'ellipsoïde de sécurité d'un segment est facilement détectable, et les coefficients de coût cp_{ji} ne sont calculés que dans le cas où l'interaction est constatée.

La fonction objective résultante pour l'optimisation inférieure est toujours l'exponentielle de la somme pondérée de ces trois critères

$$F_{\text{inf}} = e^{-(k_1 * C_1 + k_2 * C'_2 + k_3 * C'_3) / (k_1 + k_2 + k_3)}$$

Evaluation intermédiaire : Elle tient compte de deux critères. Le premier concerne la précision du suivi. Il effectue la moyenne, sur toutes les subdivisions, des valeurs d'adaptation obtenues dans la boucle inférieure :

$$T_1 = \frac{1}{n} \sum_{i=1}^n F_{\text{inf}}^i$$

Le deuxième critère fait tendre la somme des distances générées aléatoirement vers la distance désirée à parcourir ℓ_t :

$$T_2 = 1 - \frac{\sum_{i=1}^n \delta \ell_i}{\ell_t}$$

Dans le cas que nous étudions, la distance totale est $\ell_t = 4 + 0.5 + 4 = 8.5$ mètres. La fonction coût intermédiaire s'écrit donc

$$F_{\text{int}} = T_1 * e^{-k_4 * T_2}$$

Elle est optimale pour la valeur d'adaptation intermédiaire maximale $F_{\text{int}} = 1$.

Evaluation supérieure : minimiser le nombre d'intervalles divisant la trajectoire globale.

$$C_{\text{sup}} = \frac{n}{n_{\text{max}}} \quad \Rightarrow \quad F_{\text{sup}} = e^{-k_{\text{sup}} * C_{\text{sup}}}$$

avec $n_{\text{max}} = 8$ et k_{sup} fixé à 1 pour la simulation. L'évaluation supérieure des solutions utilise les valeurs de la fonction d'adaptation supérieure F_{sup} et bien entendu celles des fonction coût des étages inférieurs d'optimisation :

$$F_g = F_{\text{sup}} * F_{\text{int}}$$

où F_g est la fonction d'évaluation globale d'une solution, F_{int} celle issue de la boucle d'optimisation intermédiaire. Cette dernière ne pourra que tendre vers 1, sans l'atteindre, puisque la valeur d'adaptation supérieure sera toujours inférieure à 1 (la valeur $F_{\text{sup}} = 1$ correspond à $n = 0$, soit $X_0 = X_d$). Les conditions d'arrêt pour les étages inférieur et intermédiaire sont liées à une valeur pour la fonction coût de 1. Pour l'étage supérieur, c'est la dernière génération fixée par le paramètre évolutionnaire lié à cette optimisation qu'il faudra atteindre.

3.3.3 Opérateurs et paramètres d'évolution

Les opérateurs évolutionnaires sont identiques à ceux utilisés précédemment, à l'exception de l'opérateur de croisement :

- La **sélection** s'effectue toujours selon la méthode dite de la "roulette russe". Elle est proportionnelle à la valeur d'adaptation d'une solution, et permet de conserver les meilleurs individus tout en laissant une chance aux autres.
- Le **Croisement** est uniforme macroscopique. Il échange la valeur de deux bits entre deux parents avec une probabilité de croisement p_c .
- La **Mutation** gère l'évolution de paramètres réels ou binaires en changeant la valeur d'un bit avec une probabilité de mutation p_m .

Les paramètres évolutionnaires sont fixés par rapport aux réponses de l'algorithme. Leurs valeurs sont présentées dans le tableau 3.3. Pour l'évolution inférieure, les valeurs optimales déterminées à la section précédente sont conservées. Les paramètres liés aux boucles intermédiaire et supérieure permettent une dynamique forte tout en conservant une convergence efficace de l'évolution. L'opérateur de ré-introduction forcée n'est utilisé que pour la boucle inférieure car il n'existe a priori pas de redondance entre les solutions manipulées par les optimisation supérieure et intermédiaire.

Paramètre	Optimisation		
	Supérieure	Intermédiaire	Inférieure
k	20	100	100
g	20	100	120
p_c	0,75	1	1
p_m	0,05	0,1	0,11
p_r	1	5	5

TAB. 3.3: Paramètres évolutionnaires pour l'optimisation imbriquée

3.3.4 Résultats de simulation

Les convergences des valeurs d'adaptation pour les trois algorithmes sont illustrées par les figures 3.16, 3.17 et 3.18 ci-après. Les évolutions supérieure et intermédiaire (respectivement figures 3.18 et 3.17) sont représentées par des moyennages sur 100 simulations et l'évolution inférieure (figure 3.16) est une simulation extraite dans le cas des valeurs, pour n et $\delta\ell_i$, reportées dans le tableau 3.4. L'analyse des différentes

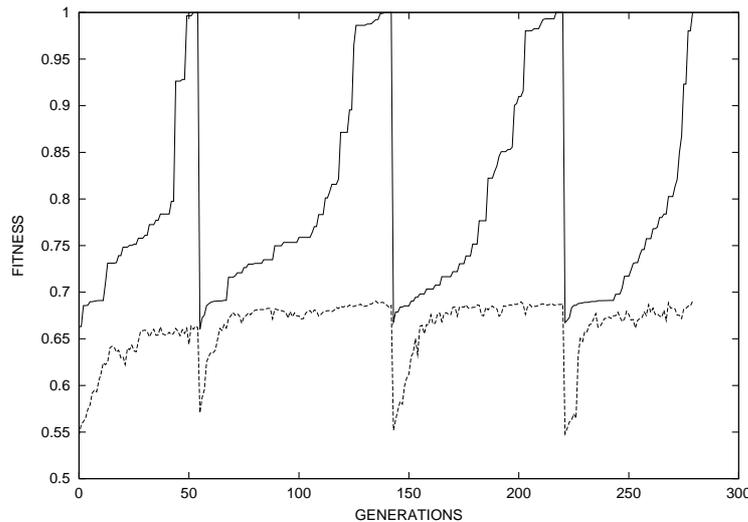


FIG. 3.16: Convergence de l'adaptation inférieure, $n = 4$ et $F_{\text{inf}} = 1$ (— Meilleur individu - - - Moyenne de la population)

évolutions a montré que le coefficient lié à l'évaluation intermédiaire, dans la fonction globale F_g , devait être mis à 2 pour que l'influence de la minimisation de n dans F_{sup} ne dépasse pas celle de la réalisation de la tâche, donnée par F_{int} . De plus, la valeur de cette dernière sort de la boucle d'optimisation intermédiaire à 1, pour tout n supérieur à 4. C'est donc la valeur de n seule qui décide des meilleures solutions, pour la répartition géographique, dès que le nombre de divisions minimal pour pouvoir réaliser le suivi est atteint. Le tableau 3.4 donne un exemple de résultat qui valide l'objectif. Comme nous l'avons vu dans la partie précédente sur l'étude de la génération de consignes, les valeurs brutes de celles-ci peuvent changer puisque aucune instruction ne concerne la vitesse de

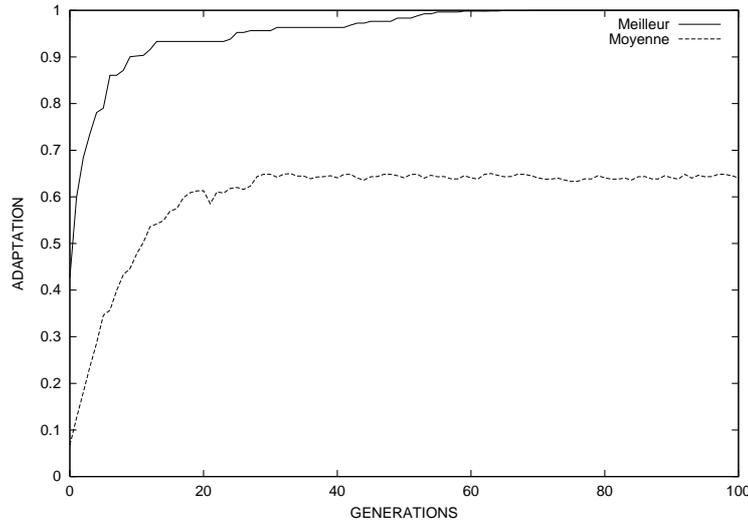


FIG. 3.17: Convergence de l'adaptation intermédiaire, moyenne sur 100 évolutions pour $n = 4$ (— Meilleur individu - - - Moyenne de la population)

Nombre de subdivisions	Répartition géographique		Entrées de Commande	
	δl_i		U_i	
$n = 4$	δl_1	3,82	U_1	(4.15; 4.15; 0.0; 0.0; 0.0)
	δl_2	0,18	U_2	(2.78; 1.04; 1.28; 0.0; 0.0)
	δl_3	0,5	U_3	(2.21; 2.94; -0.76; 0.0; 0.0)
	δl_4	4,0	U_4	(5.02; 5.02; 0.0; 0.0; 0.0)

TAB. 3.4: Solutions pour la performance de la manœuvre

réalisation de la tâche. Par contre, leurs proportions des entrées de commande les unes aux autres reste les mêmes, par exemple égalité des consignes sur les roues droite et gauche pour suivre une ligne droite. Après convergence de la répartition géographique, les valeurs obtenues pour les δl_i sont toujours les mêmes à 10^{-3} mètres près (tableau 3.4).

En ce qui concerne les graphes d'évolution, on retrouve pour l'évolution inférieure l'allure des évolutions précédemment discutées. La convergence des consignes est rapide et la valeur du paramètre g fixée à 120 est largement suffisante pour trouver la solution, lorsqu'elle existe. Lorsqu'elle n'existe pas, cette valeur basse évite de perdre trop de temps dans une recherche infructueuse. De même la valeur d'adaptation intermédiaire atteint rapidement son maximum et les 100 générations ne sont jamais toutes utilisées dès que le nombre de subdivisions dépasse 4. Dans le cas $n = 1$ (la trajectoire est alors considérée dans son ensemble), les valeurs optimales permettent à l'effecteur de parvenir à l'angle de la marche par un jeu d'entrées similaire à ceux de l'étude précédente. Une adaptation globale nulle n'est jamais obtenue, mais sa valeur minimale est obtenue dans le cas $n = 0$. La fonction d'évaluation supérieure est alors à sa valeur maximale,

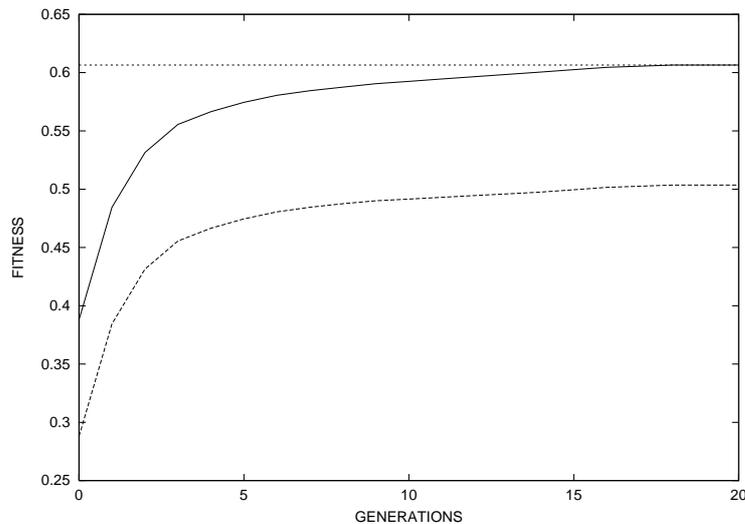


FIG. 3.18: Convergence de l'adaptation supérieure, moyenne sur 100 évolutions (— Meilleur individu - - - Moyenne de la population)

mais la solution est éliminée par la valeur très basse de l'évaluation intermédiaire.

Le temps moyen mis pour obtenir les jeux de consignes est de 9 heures 25 minutes. Ce délai est élevé malgré les convergences rapides des différents étages d'optimisation. Il est cependant possible de l'amoiner encore en supprimant un cycle d'optimisation. C'est l'objet de la partie suivante, qui propose la fusion des boucles intermédiaire et supérieure en une seule boucle d'optimisation.

3.4 Chromosomes mixtes de longueurs variables

3.4.1 Intérêt

Le gain de temps est l'intérêt principal de cette approche. Elle consiste à supprimer une boucle d'optimisation par le regroupement des cycles intermédiaire et supérieur introduits dans la partie précédente. La boucle inférieure reste indépendante pour assurer le suivi local de l'avancement de l'effecteur sur la trajectoire désirée. La boucle supérieure quantifie alors la qualité globale de la réalisation de la tâche. Les problèmes qui apparaissent sont de deux types :

- L'utilisation, *dans un même chromosome*, de valeurs réelles et entières.
- La gestion par les opérateurs évolutionnaires de *chromosomes de longueurs différentes*.

On introduit alors le concept d'un chromosome mixte (i.e. contenant des valeurs réelles et entières) dont la longueur va varier du fait même de la définition des informations qu'il contient.

3.4.2 Construction initiale des chromosomes

Les informations comprises dans ce chromosome regroupent celles des chromosomes supérieur et intermédiaire précédents, à savoir :

- Le nombre n de subdivisions qui partagent la trajectoire pour que le suivi soit réalisable dans son ensemble.
- Les dimensions $\delta\ell_i$, ($i = 1..n$), de ces intervalles géographiques.

La première information est contenue dans une valeur réelle tandis que la seconde est fournie par des entiers. Le nombre de ces réels est lui-même fonction de la valeur contenue dans l'information précédente. La génération aléatoire de ces valeurs utilise donc toujours deux opérateurs distincts (nombres réels / nombres entiers). Le chromosome

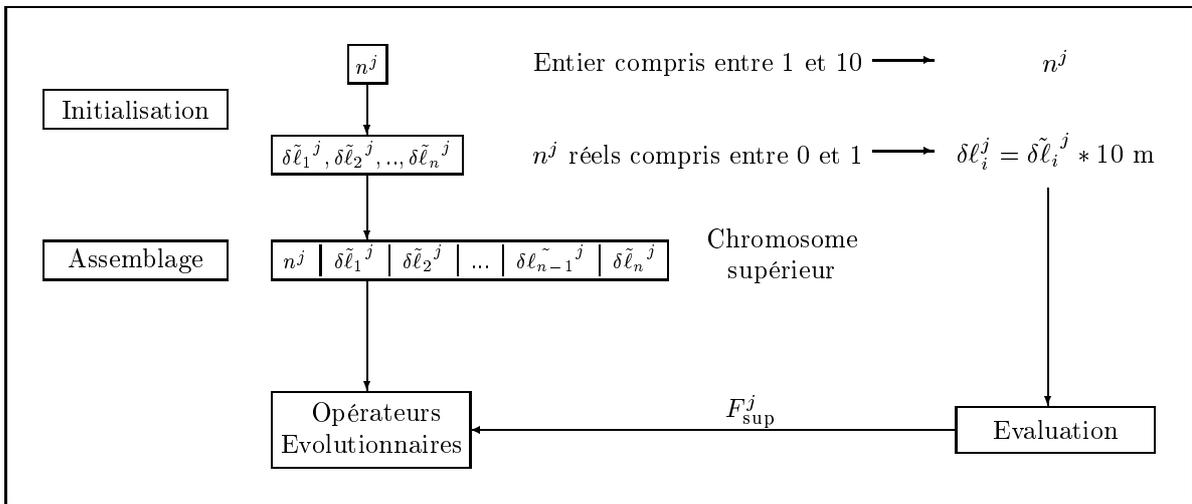


FIG. 3.19: Construction initiale des k individus supérieurs ($j = 1..k$).

est alors construit en regroupant les nombres générés aléatoirement. Sa dimension dépend toujours de la valeur du nombre de subdivision produit, elle est donc définie après génération de n . Les étapes de l'initialisation sont illustrées sur la figure 3.19. Elles peuvent être listées de la manière suivante :

1. Génération aléatoire de n , un entier compris entre 1 et 10.
2. Pour $i = 1..n$, initialisation des $\delta\tilde{\ell}_i$, réels compris entre 0 et 1.
3. Construction du chromosome supérieur par assemblage de toutes ces valeurs dans leur ordre d'initialisation. Le premier gène contient donc les informations sur la longueur du chromosome.

Ceci est effectué pour les k individus constituant la population de l'optimisation supérieure. Le décodage, en vue de l'évaluation, est effectué directement après l'initialisation des longueurs adimensionnées, leur valeur maximale étant toujours limitée à 10 mètres chacun.

La taille des individus varie entre 2 et 11 gènes ($n + 1$ gènes). La manipulation de tels chromosomes, notamment pour leur croisement, nécessite une adaptation pour éviter

les information contradictoires au sein d'un même génotype, du type $n = 5$ suivi de 8 valeurs réelles pour un même individu par exemple.

3.4.3 Manipulation des chromosomes

La lecture du premier gène de chaque chromosome est nécessaire avant toute opération sur celui-ci, car il définit sa longueur. Les différences de longueurs ne gênent alors pas pour la sélection ou la mutation des individus. Par contre, une manipulation préliminaire est à effectuer pour pouvoir effectuer l'enjambement. Celle-ci se déroule

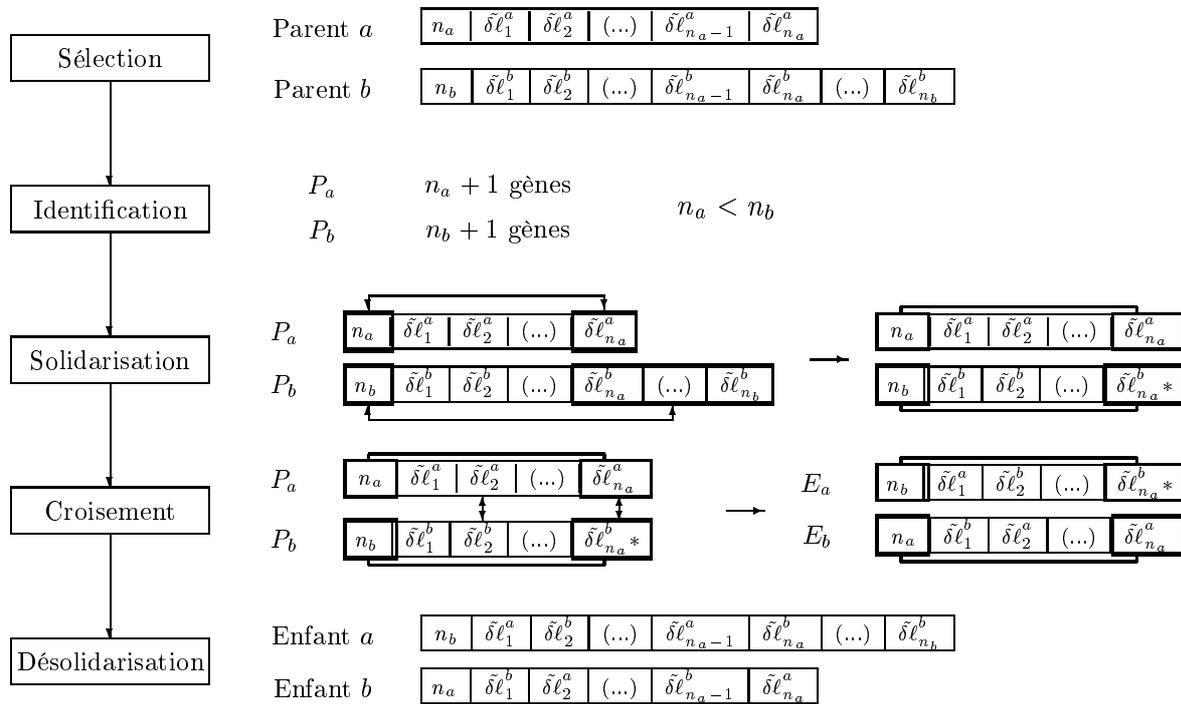


FIG. 3.20: Croisement de deux chromosomes parents a et b de longueurs différentes. Cas où $n_a < n_b$.

en 3 étapes supplémentaires par rapport aux algorithmes précédemment décrits (figure 3.20) :

1. Sélection de deux chromosomes géniteurs, notés a et b .
2. Détection de leurs longueurs $n_a + 1$ et $n_b + 1$, par lecture de n_a et n_b du premier gène de chaque chromosome. Identification du chromosome le plus long.
3. Dans le cas où n_a est inférieur à n_b (cas de la figure 3.20) : un lien virtuel est établi entre les gènes 1 et n_{a+1} du parent a d'une part, d'autre part entre les gènes 1 et $(n_a + 1)$ à $(n_b + 1)$ du parent b .
4. Le croisement peut alors s'effectuer de manière classique en considérant les gènes solidarisés comme un gène unique dans le chromosome.

5. Re-décomposition des deux chromosomes enfants. On a alors à nouveau deux chromosomes de longueur $n_a + 1$ et $n_b + 1$.

Ces manipulations préparatoires sont bien sûr inutiles lorsque les deux géniteurs sont de la même taille, et inutilisées lorsque les premières ou dernières parties des chromosomes ne sont pas échangées. Ainsi, dans le cas $n_a = n_b$, les géniteurs passent au travers des étapes de fusion et de séparation de part et d'autre de l'enjambement. A la fin de l'évolution, la taille des individus vont tendre vers les mêmes valeurs. Cette distinction $n_a = n_b$ ou $n_a \neq n_b$ permettra un gain de temps non négligeable au cours de cette manipulation.

3.4.4 Convergence rapide

La boucle d'optimisation inférieure n'est pas modifiée, les paramètres évolutionnaires inférieurs sont conservés aux mêmes valeurs que précédemment ainsi que l'expression de la fonction d'adaptation inférieure F_{inf} . Les valeurs des paramètres liés à l'optimisation supérieure sont reportés dans le tableau 3.5. Seul le nombre de générations est augmenté

Paramètre	Optimisation	
	Supérieure	Inférieure
k	100	100
g	150	120
p_c	1	1
p_m	0,11	0,11
p_r	5	5

TAB. 3.5: Paramètres évolutionnaires liés à la double optimisation

de manière à prendre en compte la complexité croissante des informations portées par les génotypes supérieurs et laisser le temps pour la convergence.

Les résultats obtenus répondent à l'objectif fixé puisque le temps mis pour la convergence vers une solution est divisé par 8, passant de 9h25 à 1h12 en moyenne sur 100 simulations, sans faire baisser la qualité de cette solution. L'évaluation supérieure est effectuée de la même manière que précédemment, elle regroupe les critères contenus dans F_g :

$$F_g = T_1 * e^{\frac{-k_4 * T_2 - k_{\text{sup}} * C_{\text{sup}}}{k_4 + k_{\text{sup}}}}$$

avec $k_4 = 2$ et $k_{\text{sup}} = 1$. Ainsi, bien que légèrement plus lente, l'évolution présente la même allure que l'évolution intermédiaire précédente (figure 3.21). La limite de convergence est celle de F_g de l'évolution supérieure précédente.

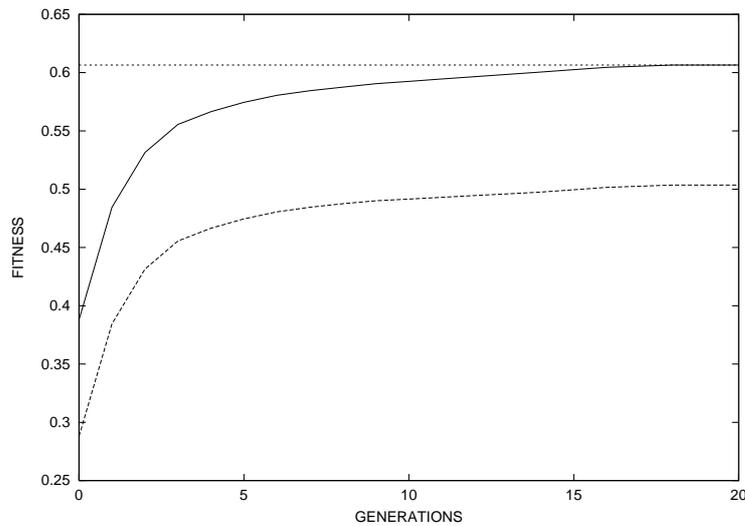


FIG. 3.21: Evolution de l'adaptation avec utilisation de chromosomes mixtes de longueur variable.

L'algorithme mis en place permet une division intuitive de la tâche (suivi de trajectoire d'un point initial à un point final désirés) en sous-tâches par l'optimisation. Les caractéristiques d'une sous-tâche sont alors définies par des entrées de commande constantes pour l'algorithme (changement de vecteur d'entrées = changement de sous-tâche). Dans l'environnement, il est difficile d'extraire une définition claire et d'établir précisément la division d'une tâche en sous-tâches caractéristiques. Un avantage supplémentaire de l'approche évolutionnaire est qu'elle ne nécessite pas un tel niveau de précision pour proposer des solutions adaptées. Une connaissance précise de la tâche globale et de l'environnement est cependant essentielle.

3.5 Conclusion

Ce chapitre était destiné à la mise en place d'un générateur de consignes fiable, utilisant en entrée le nombre de degrés de libertés du robot et une description précise des contraintes liées à la tâche et à l'environnement, en sortie les différentes entrées de commande à appliquer sur les actionneurs. L'approche évolutionnaire s'est montrée performante pour répondre à ce problème. Il ressort de cette étude que cette famille d'algorithmes constitue un outil puissant et performant pour trouver des solutions adaptées dans des problèmes de causes à effets, dont le lien entrée/sortie est mal ou méconnu. L'inconvénient majeur de ces algorithmes est qu'ils nécessitent une mise en place importante, tant du point de vue méthodologie que logiciel, et impliquent une pose du problème adaptée. Le temps de mise en place est donc très important, d'autant plus avec la complexité du problème qui se traduit souvent au niveau de l'évaluation. Celle-ci, bien que pouvant utiliser une fonction d'adaptation irrégulière, doit être simplifiée et minimisée au maximum sous réserve de voir apparaître des réponses pour le moins surprenantes au problème posée. Cette simplicité ne suffit d'ailleurs pas toujours,

comme nous l'avons constaté, pour assurer sa validité. Beaucoup de temps de mise en place, donc, mais lorsque la phase d'adaptation est passée, l'efficacité de la méthode est très grande.

	Adaptation	Inconvénients	Amélioration	Convergence
Trajectoire simple	Basique	Emergence rare, disparition des bonnes solutions	-	$F_{moy} = 0.7$ 285 générations min.
	Critère	Emergence irrégulière, disparition des bonnes solutions	Emergence plus fréquente 18% → 10% d'échecs	$F_{moy} = 0.71$ 250 générations min.
	Récupération	Emergence non systématique	Emergence plus fréquente et conservation des bonnes solutions	$F_{moy} = 0.84$ $g_{min} = 187$ $g_{max} = 280$
	Paramètres		Emergence systématique et plus rapide	$F_{moy} = 1$ 120 générations max.
Manœuvre simple	Optim. triple	Temps : 9h25 en moyenne		systématique sur les trois niveaux
	Optim. double		1h12 en moyenne	systématique sur les deux niveaux

TAB. 3.6: Récapitulation des différentes adaptations de l'approche évolutionnaire pour le problème de la génération de consignes.

Le tableau 3.6 est un récapitulatif des différentes adaptations effectuées au cours de ce chapitre, ainsi que des améliorations qui en découlent, dans le cas de trajectoires sans puis avec manœuvre. L'approche de notre méthodologie est basée sur le fait qu'un suivi de trajectoire par l'organe terminal du manipulateur mobile peut résulter de la succession de vecteurs d'entrées de commande constants appliqués sur les 5 actionneurs de notre robot pendant un laps de temps défini par l'optimisation. Ainsi, le cas sans manœuvre correspond à l'application d'un unique vecteur de consignes constant. Les performances de l'algorithme, dans ce cas, ont été étudiées de manière à être systématiques et rapides (tableau 3.6). Le temps d'application de ces entrées est le temps nécessaire pour parvenir au point final de la trajectoire. La rapidité de réalisation de la trajectoire n'est pas prise en compte bien qu'un délai maximal de 5 minutes soit accordé pour la réalisation de la tâche. L'évaluation des solutions est alors basée sur l'erreur en position (écart à la trajectoire désirée), l'avancement sur la trajectoire et l'évitement de collision. Le cas avec manœuvre utilise les résultats précédents

comme optimisation inférieure qui permet un retour local sur l'avancement de l'effecteur. Une évaluation globale est alors mise en place sur deux boucles d'optimisation distinctes qui permettent de minimiser le nombre de consignes différentes à appliquer et le temps d'application de ces consignes. Ces deux boucles ont été réduites à une seule de manière à diminuer le temps de simulation, par la construction et l'utilisation de génotypes mixtes entier/réels dont la longueur varie en fonction des informations qu'ils contiennent. Toutes ces adaptations ont permis la validation d'un algorithme évolutionnaire à deux boucles d'optimisation, permettant l'émergence systématique d'une solution adaptée en 1h12mn en moyenne.

Nous avons donc développé une commande qui utilise uniquement le nombre de ddl d'une structure pour effectuer une tâche de suivi de trajectoire. Elle a été testée avec succès dans le cas d'une structure redondante, manipulateur mobile constitué d'éléments à paramètres dynamiques connus et fixés. Les contraintes ont été conservées à un niveau minimal (par leur nombre et leur expression), de manière à obtenir une évaluation objective et fiable des solutions proposées par l'algorithme évolutionnaire. Le chapitre suivant met en place les mécanismes qui recevront ces consignes par la suite.

Chapitre 4

Élimination préliminaire des structures

4.1 Position du problème

4.1.1 Objectif

Notre approche générale propose la décomposition de la synthèse de manipulateurs mobiles en deux étapes distinctes :

- une première phase d'élimination des structures mécaniques incompatibles avec la réalisation de la tâche ;
- La synthèse finale effectuée sur les structures restantes de la première phase en tenant compte du déplacement du robot dans l'évaluation globale du système.

Ce chapitre est destiné à l'étude et la mise en place de la première phase. Son rôle est de limiter la synthèse finale aux seules structures pouvant potentiellement réaliser la tâche de manière à accélérer le processus global de conception. Dans cet objectif, la méthodologie doit répondre à différents problèmes :

- Éliminer les structures incompatibles avec la réalisation de la tâche tout en conservant un maximum de diversité dans les structures restantes, c'est-à-dire trouver un équilibre entre la rigueur de l'évaluation (sélectivité) et la conservation de la diversité des solutions retenues ;
- Simplifier l'étude par la détermination d'un ensemble minimaliste de variables à déterminer tout en conservant leur validité ;
- Simplifier l'étude par la décomposition de la tâche en sous-tâches caractéristiques tout en conservant la validité des solutions retenues par rapport à la tâche globale.

Bien que cette première étude soit simplifiée (ensemble minimaliste de variables à déterminer, étude statique), elle nécessite l'utilisation d'une méthode d'optimisation globale du fait de l'impossibilité d'évaluer ces variables de conception séparément [Chedmail98]. En effet, une approche déterministe ou par composition ne serait que mal adaptée au problème. La première du fait de l'irrégularité de la fonction objective construite

pour le problème et de la dimension très grande de l'espace de recherche des solutions [Arora95]. La seconde du fait du couplage entre ces variables pour une résolution en sous-problèmes d'optimisation [Ferber95].

Le problème est donc traité dans toute sa complexité. Cette approche d'optimisation globale a été associée à la technologie évolutionnaire dans le cadre de la synthèse géométrique des robots manipulateurs [Chedmail96] [Chen96] [Chocron00]. Nous allons en retracer les différentes étapes tout en les adaptant à notre problème particulier.

4.1.2 Décomposition de la tâche

La tâche globale d'un manipulateur mobile revient à suivre l'évolution de la situation de l'effecteur dans le temps -trajectoire en position et orientation- ainsi que celle de l'effort applicable par l'effecteur. Elle peut donc être définie, à l'effecteur, par (Fig. 4.1) :

- le suivi de trajectoire $X(t)$ entre les positions initiale X_i et finale X_f .
- L'orientation $\phi(t)$ évoluant entre les orientations initiale ϕ_i et finale ϕ_f .
- L'effort $F(t)$.

La réalisation de la tâche consiste à faire coïncider les variables $X(t)$, $\phi(t)$ et $F(t)$ avec $X_d(t)$, $\phi_d(t)$ et $F_d(t)$, respectivement la position, l'orientation et l'effort désirés à l'effecteur à l'instant t . Elle s'accompagne des contraintes d'environnement qui concernent l'ensemble du robot : l'évitement de collision interne ou avec un obstacle (mur, pilone), le sol.

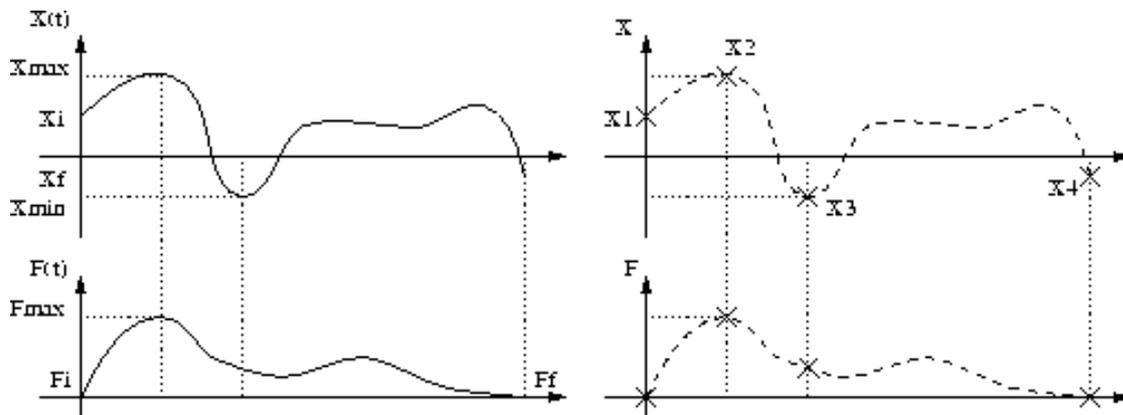


FIG. 4.1: Extraction de quatre sous-tâches caractéristiques d'une tâche globale.

La tâche globale peut être partiellement caractérisée par des situations *instantanées* extraites de la trajectoire et les efforts instantanés correspondants. Ainsi, la validation d'une tâche passe par le respect de n sous-tâches définies aux instants t_i ($i = 1, \dots, n$) par :

- une situation (position + orientation),
- un effort,
- des contraintes d’environnement.

La différenciation des sous-tâches se fait explicitement par la modification d’un de ces paramètres caractéristiques. La réciproque ne s’applique pas puisque le fait qu’un robot parvienne à réaliser les n sous-tâches ne signifie pas qu’il pourra s’acquitter de la tâche globale. Les sous-tâches extraites se doivent d’en être représentatives, et la pertinence de la sélection est une difficulté majeure dans la décomposition, qui augmente très rapidement avec la complexité globale. Les conditions initiales et finales, les maxima et minima de chacune des fonctions font automatiquement partie des situations à prendre en compte. D’autres sous-tâches intermédiaires peuvent être introduites dans le cas de trajectoires complexes par exemple, ou d’efforts intensifs localisés. La figure 4.1 montre un exemple de décomposition d’une tâche globale (à gauche) en quatre sous-tâches caractéristiques (à droite). Les graphes supérieurs et inférieurs représentent respectivement la position et l’effort désirés à l’effecteur. On peut alors considérer une solution robotique *potentiellement* apte à réaliser la tâche globale si elle parvient à s’acquitter de l’ensemble de ces quatre sous-tâches.

L’étude préliminaire peut donc être réalisée en statique du fait de la possibilité d’évaluer complètement les manipulateurs mobiles grâce à des critères mathématiques locaux (critères qui peuvent être géométriques, cinématiques ou dynamiques). Cette étude revient à effectuer une synthèse classique de bras manipulateurs dont la configuration de la base mobile n’est pas fixée. Chacunes des structures proposées doit être en mesure de réaliser l’ensemble des sous-tâches définies. Les solutions retenues suite à cette sélection pourront être testées par la commande dynamique pour s’assurer de la réalisation complète de la tâche. Cette étude préliminaire revient donc à effectuer une recherche de faisabilité de la tâche, qui va considérablement alléger les temps de calculs en éliminant directement de la sélection les solutions incompatibles avec les contraintes fixées, c’est-à-dire qui ne peuvent pas répondre à une ou plusieurs exigences locales. Les hypothèses que nous posons pour cadrer la recherche, les paramètres fixés ainsi que ceux à déterminer sont présentés ci-après.

4.1.3 Limites de l’étude : paramètres fixés

Nous cherchons ici à déterminer les structures cinématiques aptes à atteindre une situation donnée avec un effort donné, de manière à éliminer de la simulation celles non adaptées : chaînes cinématiques trop courtes, efforts non réalisables dans la direction désirée,... Dans cet objectif, les paramètres fixés et ceux à déterminer doivent être clairement définis. Dans notre approche, les mouvements du robot ne sont pas considérés. La base mobile est donc associée à un socle posé sur le sol, dont les dimensions (L_b, l_b, h_b) sont fixées arbitrairement à $(1; 0.8; 0.4)$ mètres, qui correspondent à celles d’une plate-forme de taille moyenne. L’environnement est constitué par un sol plat horizontal. La position de la plate-forme mobile sur le sol, et son orientation autour de l’axe vertical peuvent varier. L’obstacle à éviter est un mur modélisé par un volume

interdit. Le manipulateur mobile est donc constitué par :

- une plate-forme de dimensions fixées à positionner et orienter sur le plan défini par le sol,
- une chaîne cinématique, dont le ddl n'excède pas 6.

Le nombre de ddl maximal fixé autorise, indépendamment de la base mobile, un placement et une orientation dans l'espace. Le degré de redondance maximal correspondant est donc de 3. Outre ces limites, nous fixons aussi les hypothèses suivantes :

- La position du bras sur la base est fixée au centre de sa surface supérieure.
- L'étude est statique.

Ces hypothèses ont plusieurs objectifs : réduire les temps de calcul en simplifiant le problème, valider l'approche, fixer dès le départ les variables valuées. Dans le cadre de l'évaluation de la plate-forme mobile, il n'existe pas vraiment de critère d'évaluation (contrairement aux bras manipulateurs). Le mode de locomotion (à roues, à pattes, mixte, à chenille...) ainsi que la plupart des caractéristiques sont fixées arbitrairement. Leurs dimensions sont ensuite optimisées (énergie, manœuvrabilité...). L'évaluation objective d'une plate-forme mobile n'est pas envisageable sans tenir compte de la commande du véhicule.

4.1.4 Paramètres non valués

L'ensemble des variables de conception à déterminer se trouve donc allégé, au vu des simplifications effectuées dans cette étape préliminaire. Celui-ci contient :

- Des **paramètres topologiques** :
 - *Cinématiques* : le nombre de liaisons total de la structure, leur type (rotoïde, prismatique, fixe), leur axe $(\vec{x}, \vec{y}, \vec{z})$.
 - *Géométriques* : la dimension des constituants de la chaîne cinématique.
- Des **paramètres configurationnels**, et qui doivent prendre en compte les contraintes liées à la tâche : les valeurs articulaires à chaque liaison, la situation de la plate-forme mobile, l'effort effectif à l'effecteur.
- Des **paramètres mathématiques** : la valeur des performances pour chaque critères, de la fonction objective liée à chaque sous-tâche.

A chaque ensemble de variables topologiques correspond un ensemble de variables articulaires permettant le positionnement correct de l'organe terminal ainsi qu'une configuration respectant -le cas échéant- les contraintes extérieures.

4.2 Conception préliminaire

4.2.1 Evolutions imbriquées

L'algorithme doit donc gérer simultanément deux optimisations différentes mais non indépendantes puisque la recherche s'effectue sur la topologie du robot associée à une configuration. L'évaluation globale d'une solution concerne cette association structure/configuration.

La figure 4.2 met en évidence la décomposition du problème structure/configuration ainsi que l'impossibilité de les dissocier du point de vue de l'évaluation des solutions. Chacune des étapes figurant sur ce schéma va être détaillée dans les parties suivantes. Ces étapes concernent :

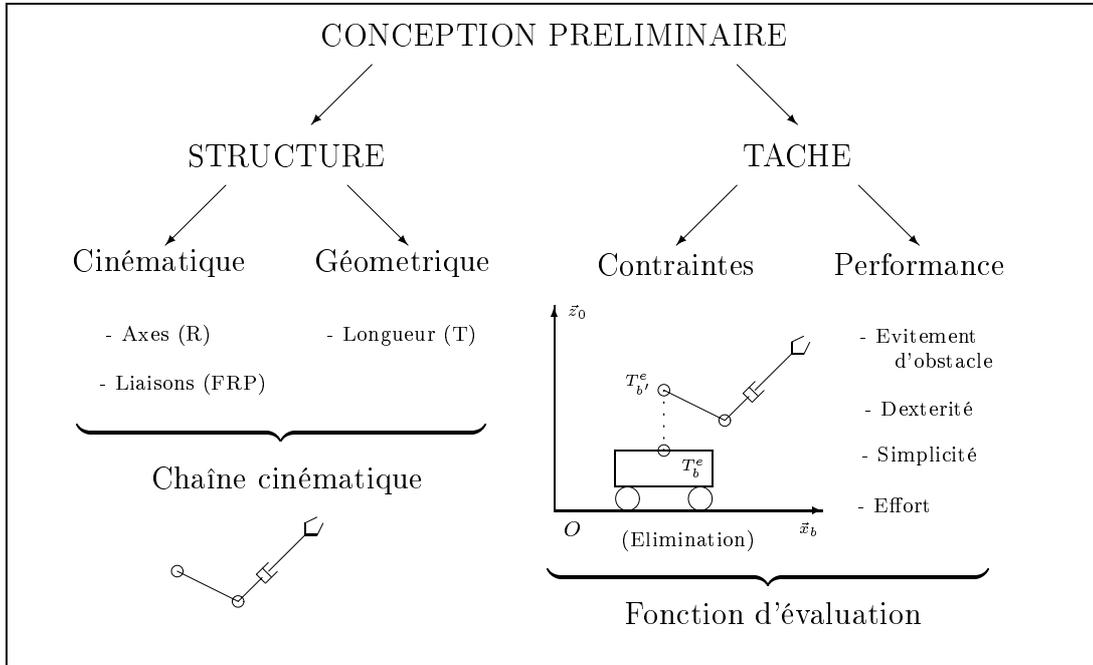


FIG. 4.2: Décomposition de la conception préliminaire.

1. La construction d'une solution topologique répondant à des règles précises d'assemblage.
2. L'écriture du modèle de cette structure qui permet la quantification du respect des contraintes géométriques liées au robot et à la sous-tâche.
3. L'introduction d'une configuration dans le modèle.
4. L'évaluation de l'ensemble (topologie + configuration) par des critères qui tiennent compte des contraintes liées
 - à la sous-tâche : atteindre une situation, appliquer un effort.
 - à l'environnement : éviter les obstacles.
 - à la technologie : minimiser la taille du robot (longueur des barres, ddl), éviter les configurations singulières.

Chaque étape inclue des tests d'élimination stricts dans le cas où une des contraintes dures (liées uniquement à la réalisation de la sous-tâche) ne serait pas respectée.

4.2.2 Cinématique modulaire

La cinématique modulaire que nous utilisons est inspirée, pour sa simplicité, du concept TOMMS [Matsumaru95]. Les robots sont construits par un assemblage de 3

modules différents, de géométries simples, qui permettent d'accéder à une infinité de possibilités. Ceux-ci sont assemblés aléatoirement de manière à obtenir une structure cinématique complète (Fig. 4.3). Leurs caractéristiques sont regroupées dans le tableau

Modules	Type	ddl	Transformation
Base	Structurel	0	translation suivant \vec{z}
Barre	Structurel	0	translation suivant \vec{x}
Pivot	Articulaire	1	rotation selon \vec{z}
Glissière	Articulaire	1	translation selon \vec{x}
Fixe	Articulaire	0	Identité

TAB. 4.1: Modules mécatroniques pour la cinématique modulaire.

4.1. Ils sont de type structurel (base, barres) ou articulaire (liaisons). Ces modules

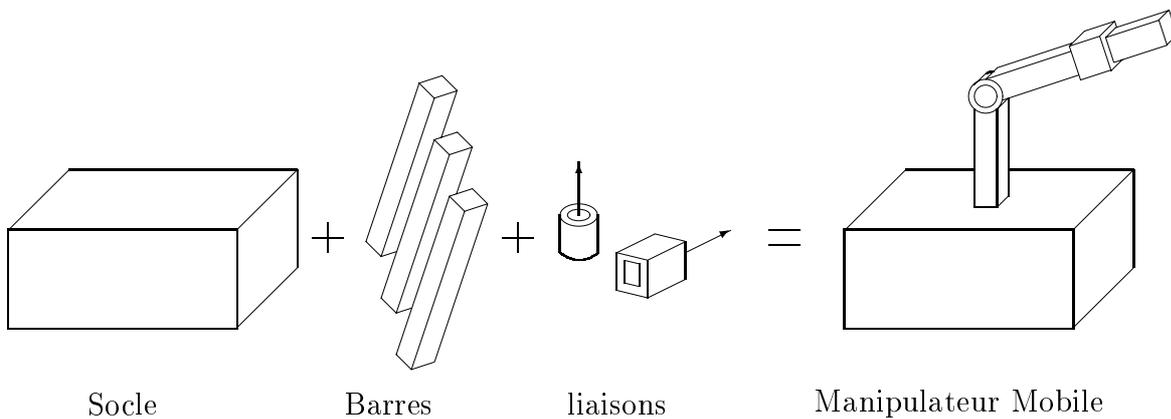


FIG. 4.3: Assemblage de 3 modules différents pour la constitution d'un robot.

sont :

- Un parallélépipède représentant l'enveloppe de sécurité autour de la plate-forme mobile introduite au chapitre 2.
- Un ensemble de barres, constituant de la chaîne cinématique. Leur longueur initiale est déterminée aléatoirement et varie entre 0 et 1 mètre.
- Un ensemble de liaisons dont la quantité n'excède pas le nombre de barres et dont les caractéristiques (nature et axe) sont fixées aléatoirement.

La chaîne cinématique qui modélise le manipulateur est représentée par un assemblage de segments, chaque segment comprenant une liaison et une barre. Ceux-ci sont donc complètement caractérisés par

- le type de la liaison (rotoïde, prismatique ou fixe),
- son orientation (axe \vec{x}_i , \vec{y}_i ou \vec{z}_i pour le $i^{\text{ème}}$ segment),
- la longueur de la barre (entre 0 et 1 mètre).

Pour atteindre une situation quelconque, un manipulateur doit comporter 6 ddl, soit 6 segments. La représentation de la chaîne cinématique permet de définir complètement

le manipulateur et d'en déduire le modèle géométrique direct dans le repère qui lui est lié, à savoir le repère de la base.

4.2.3 Modélisation et contraintes géométriques

La cinématique modulaire utilisée (figure 4.3) est conceptuellement simple et permet d'accéder à une large diversité de topologies cinématiques. A chaque segment i ($i = 1..n_b$) est associée une matrice homogène \mathbf{T}_i^{i+1} telle que

$$\mathbf{T}_i^{i+1}(O = [\emptyset/\vec{x}/\vec{y}/\vec{z}], T = [R/P/F], L = [0..1]) = R_O^{90} * \mathbf{T}_T^{q_{i+1}} * t_x^L$$

$$\text{avec} \quad \mathbf{T}_T^{q_{i+1}} = \begin{cases} R_z^{q_{i+1}} & : \text{ si } T = R \text{ avec } q_{i+1} = 2\pi * \tilde{q}_{i+1} \\ t_x^{q_{i+1}} & : \text{ si } T = P \text{ avec } q_{i+1} = L * \tilde{q}_{i+1} \\ \mathbb{I} & : \text{ si } T = F \end{cases}$$

L'équation de structure de la chaîne cinématique constituée par l'assemblage des segments peut donc s'écrire :

$$\mathbf{T}_0^e = \prod_{i=0}^{n_b} \mathbf{T}_i^{i+1}$$

- \mathbf{T}_0^e : Matrice homogène de configuration de l'organe terminal
- \mathbf{T}_i^{i+1} : Matrice homogène de transformation pour le segment $i + 1$
- R_α^β : Matrice homogène de rotation d'axe α et d'angle β
- t_x^γ : Matrice homogène de translation d'axe \vec{x} et de longueur γ

Les structures agencées sont modélisées par l'utilisation des matrices homogènes associées à chaque segment. Les notations utilisées dans cette partie et pour toute la suite sont présentées sur la figure 4.4. D'une manière générale, l'indice "0" correspond à l'environnement, "b" à la plate-forme mobile, "e" à l'organe terminal de la chaîne cinématique (l'effecteur) et "d" à l'objectif (la situation désirée de l'effecteur) (figure 4.4). Le point $B(x_b, y_b, h_b)$ correspond à l'emplacement de la chaîne cinématique sur la plate-forme, au centre de sa face supérieure.

La situation de l'effecteur dans le repère de référence \mathcal{R}_0 s'exprime par la matrice de transformation homogène T_0^e telle que

$$T_0^e = T_0^{b'} T_{b'}^e$$

où $\mathcal{R}_{b'}$ est le repère lié à la base du bras manipulateur (Fig. 4.4). Lorsque la plate-forme se déplace sur le sol, sa matrice de transformation associée correspond à une rotation autour de l'axe vertical \vec{z}_0 et à la position au point B :

$$T_0^b = \begin{pmatrix} \cos \theta & \sin \theta & 0 & x_b \\ -\sin \theta & \cos \theta & 0 & y_b \\ 0 & 0 & 1 & h_b \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

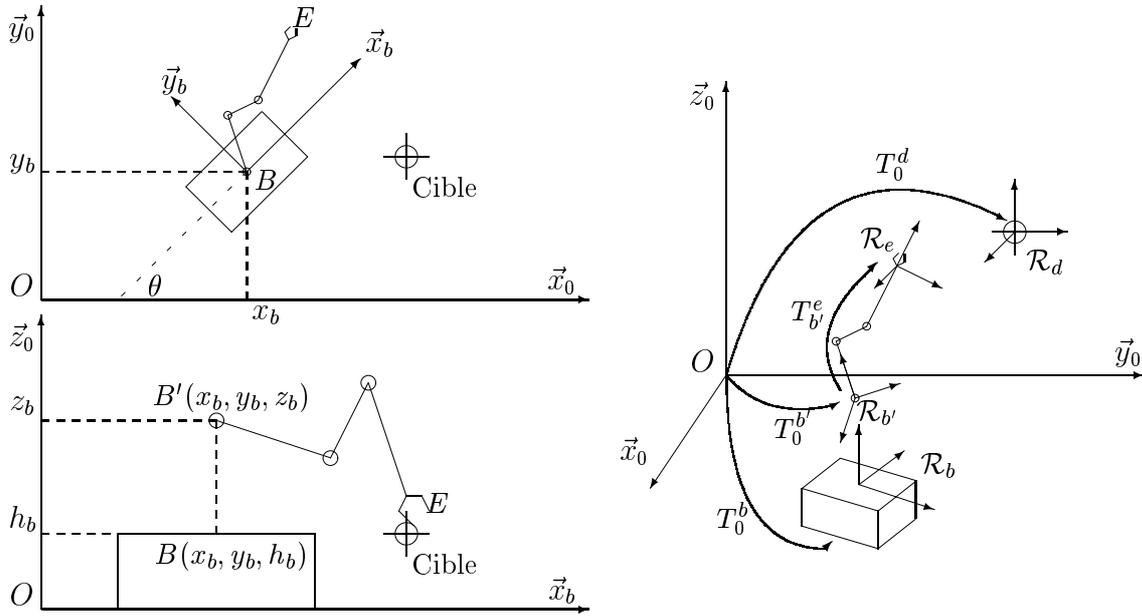


FIG. 4.4: Notations.

où $\theta(t) = (\vec{x}_0, \vec{x}_b)$ est l'orientation de la base dans le repère de référence. Lorsque le bras manipulateur représenté par la chaîne cinématique est posé sur la plate-forme mobile, les deux repères \mathcal{R}_b et $\mathcal{R}_{b'}$ sont superposés. La matrice de transformation $T_0^{b'}$ est alors de la forme de T_0^b . On a donc

$$B' = B \quad \Rightarrow \quad T_0^{b'} = T_0^b \quad \Rightarrow \quad T_0^{e'} = T_0^e$$

où T_0^e est la matrice de transformation du repère de la plate-forme \mathcal{R}_b au repère lié à l'effecteur \mathcal{R}_e . Le Modèle Géométrique Direct de la chaîne cinématique générée par l'algorithme fournit directement la matrice T_0^e . La position de la base (x_b, y_b) doit être déterminée de telle sorte que la cible soit dans l'espace de travail du manipulateur. Plusieurs possibilités s'offrent pour obtenir cette situation : on peut demander à l'algorithme de tester une situation quelconque et de converger vers les solutions qui nous intéressent. L'espace de travail peut être évalué pour chaque manipulateur généré de manière à proposer directement des situations convenables. Nous optons pour une autre approche qui consiste à considérer que l'effecteur a bien atteint la cible. On a alors

$$T_0^e = T_0^d$$

où T_0^d est la matrice de transformation désirée, de \mathcal{R}_0 au repère lié à la cible \mathcal{R}_d . La matrice $T_0^{b'}$ que nous désirons faire tendre vers T_0^b s'obtient alors directement :

$$T_0^{b'} = T_0^d (T_0^e)^{-1}$$

Cette méthode évite d'évaluer l'espace de travail du manipulateur et d'envisager des solutions en dehors de cet espace, donc économise du temps de calcul.

L'optimisation doit donc avoir pour issue, entre autres, une matrice $T_0^{b'}$ dont la forme se rapproche de celle de T_0^b . Les critères géométriques établis doivent prendre en compte cette contrainte pour établir la faisabilité en situation pour la sous-tâche considérée.

4.2.4 Evaluation des solutions

Dans le cadre de l'évaluation des solutions proposées, nous désirons laisser un maximum de liberté à l'algorithme de manière à obtenir une grande variété de chaînes cinématiques. Celles-ci doivent toutefois réaliser les contraintes géométriques (situations de ses deux extrémités imposées) et de l'environnement (évitement d'obstacles). Du point de vue de la structure, les bras les plus légers seront favorisés dans l'objectif d'être embarqués sur une plate-forme. Autre paramètre important, la mobilité de l'effecteur sera évaluée de manière à éviter les configurations singulières handicapantes pour les déplacements de l'effecteur autour de la position désirée.

Contraintes géométriques

Comme nous l'avons vu dans le paragraphe précédent, les repères outils et tâche sont considérés superposés, i.e. l'extrémité supérieure de la chaîne cinématique coïncide avec la cible. La génération aléatoire des configurations permet de déterminer la position et l'orientation correspondantes de l'extrémité inférieure de la chaîne cinématique, décrivant ainsi l'ensemble du domaine accessible de celle-ci sans que nous avons besoin de le déterminer explicitement. La matrice $T_0^{b'}$ déduite du MGD est de la forme

$$T_0^{b'} = \begin{pmatrix} (T_0^{b'})_{11} & (T_0^{b'})_{12} & (T_0^{b'})_{13} & (T_0^{b'})_{14} \\ (T_0^{b'})_{21} & (T_0^{b'})_{22} & (T_0^{b'})_{23} & (T_0^{b'})_{24} \\ (T_0^{b'})_{31} & (T_0^{b'})_{32} & (T_0^{b'})_{33} & (T_0^{b'})_{34} \\ (T_0^{b'})_{41} & (T_0^{b'})_{42} & (T_0^{b'})_{43} & (T_0^{b'})_{44} \end{pmatrix} = \begin{pmatrix} (T_0^{b'})_{11} & (T_0^{b'})_{12} & (T_0^{b'})_{13} & x_b \\ (T_0^{b'})_{21} & (T_0^{b'})_{22} & (T_0^{b'})_{23} & y_b \\ (T_0^{b'})_{31} & (T_0^{b'})_{32} & (T_0^{b'})_{33} & z_b \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

La position de la plate-forme (x_b, y_b) dans le plan horizontal est donc obtenue directement par déduction des termes de la troisième colonne de la matrice :

$$\begin{aligned} x_b &= (T_0^{b'})_{14} \\ y_b &= (T_0^{b'})_{24} \end{aligned}$$

Nous éliminons alors les solutions incompatibles avec la géométrie du socle, c'est-à-dire les chaînes ne permettant pas d'atteindre la cible lorsque le bras est fixé sur la base. Cette situation est réalisée lorsqu'il y a égalité entre les deux matrices de transformation

$$\begin{aligned} T_0^{b'} &= T_0^b \\ \Rightarrow \begin{pmatrix} (T_0^{b'})_{11} & (T_0^{b'})_{12} & (T_0^{b'})_{13} & x_b \\ (T_0^{b'})_{21} & (T_0^{b'})_{22} & (T_0^{b'})_{23} & y_b \\ (T_0^{b'})_{31} & (T_0^{b'})_{32} & (T_0^{b'})_{33} & z_b \\ 0 & 0 & 0 & 1 \end{pmatrix} &= \begin{pmatrix} \cos \theta & \sin \theta & 0 & x_b \\ -\sin \theta & \cos \theta & 0 & y_b \\ 0 & 0 & 1 & h_b \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{aligned}$$

où θ est l'orientation de la plate-forme autour de l'axe vertical. Sa valeur est obtenue après convergence :

$$\theta = \cos^{-1}(T_0^{b'})_{11}$$

De la forme de la matrice T_0^b on peut déduire les trois critères permettant le respect des contraintes géométriques du système : l'altitude z_b doit tendre vers la hauteur de la plate-forme, soit

$$L = 1 - \frac{|(T_0^{b'})_{34}|}{h_b}$$

Deux conditions sont nécessaires et suffisantes pour vérifier que la forme de la matrice de rotation déduite du MGD tende vers celle d'une rotation d'axe \vec{z} :

$$A_1 = 1 - \sum_{j=1}^3 (T_0^{b'})_{1j}^2$$

$$A_2 = 1 - \sum_{i=1}^3 (T_0^{b'})_{i2}^2$$

où i et j représentent respectivement les indices de lignes et de colonnes de la matrice $T_0^{b'}$. Ces trois critères doivent être minimisés, leur valeur optimale étant à 0.

Evitement d'obstacles

La méthode de l'ellipsoïde décrite au chapitre précédent est à nouveau utilisée. Le critère d'évitement de collision s'écrit

$$O = \sum_{j=0}^{n_b} \sum_{i=1}^{n_o} cp_{ji}$$

avec

$$\text{Si } \text{Obstacle}_i \in ES_j \Rightarrow cp_{ji} = 0$$

$$\text{Si } \text{Obstacle}_i \in ES_j \Rightarrow cp_{ji} = \frac{(d_s - d_{ji})}{d_s}$$

où ES_j est l'Ellipsoïde de Sécurité autour du segment j , cp_{ji} le coefficient de coût lié à la $j^{\text{ème}}$ barre du manipulateur ou à la base, d_s la distance de sécurité à l'obstacle et d_{ji} la distance de l'élément considéré à l'obstacle (figure 3.15). Si aucun obstacle n'est au contact de cette zone, le coût est à 0. Dès qu'un obstacle pénètre dans l'ellipsoïde de sécurité, le coût augmente jusqu'à sa valeur maximale, 1, qui traduit le contact à l'obstacle. En plus des obstacles de l'environnement, le socle et le sol sont définis comme des obstacles permanents au bras manipulateur, de manière à éviter les collisions internes du robot.

Poids

Le poids du robot affecte sa dynamique générale. Nous allons donc favoriser les manipulateurs minimaux qui pourront réaliser la tâche. L'approche modulaire utilise des barres identiques pour toute la chaîne cinématique, dont seule la longueur varie. On peut donc assimiler le robot le moins lourd à celui dont la longueur des barres est la moins longue :

$$P = \frac{\sum_{i=1}^{n_b} l_i}{l_{\max}}$$

où n_b est le nombre de ddl du bras manipulateur, l_i la longueur de la $i^{\text{ème}}$ barre, l_{\max} la longueur maximale du manipulateur. l_{\max} dépend du nombre de ddl du manipulateur, chaque barre mesure au maximum 1 mètre. Donc $l_{\max} = n_b$ mètres. Ce critère doit aussi être minimisé et est majoré par 1, mais au contraire des précédents critères, il ne peut bien évidemment pas être annulé. La simulation va tester des robots comprenant 1 à 6 ddl et retiendra essentiellement ceux dont la structure est minimale pour réaliser la tâche. Le critère de poids présente l'avantage de ne pas dépendre du nombre de ddl du manipulateur, puisqu'il effectue une moyenne sur la longueur des barres. Ceci permet de ne pas écarter prématurément les manipulateurs à nombre de ddl supérieur à ceux nécessaires pour réaliser la tâche. Par la suite, l'adaptabilité du manipulateur mobile final à s'acquitter d'une nouvelle tâche par rapport à celle prévue initialement s'en trouvera améliorée.

Mobilité

L'évaluation de la mobilité est un problème délicat dans le sens où le critère doit pouvoir s'appliquer à n'importe quelle chaîne cinématique tout en restant simple de mise en place. La manipulabilité cinématique ω introduite par Yoshikawa par exemple ne peut pas être utilisée puisqu'elle ne concerne que les manipulateurs rotoïdes d'une part, d'autre part utilise la jacobienne des vitesses qui suppose l'expression du modèle cinématique du robot.

Pour évaluer la mobilité du système à son effecteur, on estime le torseur des efforts *théoriques* maximaux que celui-ci peut produire du fait de ses moteurs. Pour cela nous considérons que les moteurs peuvent fournir un couple maximal C_{\max} pour les liaisons rotoïdes et une force maximale F_{\max} pour les liaisons prismatiques. Le torseur statique exprimé à l'effecteur fournit l'effort recherché et permet de déterminer dans quelle direction les articulations de la chaîne cinématique autorise le déplacement de l'effecteur. La valeur de cet effort doit être maximal dans toutes les directions. Le critère d'effort adimensionné C_f s'écrit :

$$C_f = \frac{\|\vec{F}_{\max} - \vec{F}_e\|}{\|\vec{F}_{\max}\|}$$

- \vec{F}_{\max} : effort maximal à l'effecteur
- \vec{F}_e : effort théorique productible par l'effecteur

Le torseur statique fournit une autre information importante pour notre système : si une valeur de coordonnée d'un de ces deux vecteurs est nulle, la mobilité l'est aussi dans ou autour de la direction correspondante. La force \vec{F}_{max} est un effort de référence qui dépend du nombre de ddl du bras manipulateur considéré.

Fonction objective globale

La fonction d'adaptation globale F est la concaténation de deux fonctions distinctes aux objectifs différents. Elles correspondent aux évaluations inférieure et supérieure, respectivement F_{config} et F_2 , telles que

$$F_{\text{config}} = \exp\left(-\frac{k_L * L + k_{A_1} * A_1 + k_{A_2} * A_2 + k_O * O}{k_L + k_{A_1} + k_{A_2} + k_O}\right)$$

$$F_2 = \exp\left(-\frac{k_P * P + k_f * C_f}{k_P + k_f + k_L + k_{A_1} + k_{A_2} + k_O}\right)$$

F_{config} est donc une évaluation locale des solution et concerne la réalisation brute de la tâche. Lorsque $F_{\text{config}} = 1$, l'effecteur atteint la cible sans toucher les obstacles. F_2 est une évaluation globale du système et concerne en plus la performance dans la réalisation de la tâche. Elle propose un compromis entre la taille du robot et la dextérité autour de la cible. La fonction globale résultante F s'écrit donc

$$F = \tilde{F}_{\text{config}} * F_2$$

où $\tilde{F}_{\text{config}}$ est la valeur d'adaptation inférieure F_{config} moyennée sur toutes les sous-tâches. Une solution topologique est automatiquement retenue lorsque $F_{\text{config}} = 1$ et pour la meilleure valeur de F_2 . La distinction entre ces deux fonctions est utile lors de la sauvegarde des solutions topologiquement aptes.

Tous les critères sont positifs, la fonction globale est donc majorée par 1. La meilleure valeur de cette fonction ne peut cependant pas atteindre sa valeur maximale, le critère de poids ne peut être annulé. Mais elle peut se rapprocher de 0. Ainsi, si tous les critères sont remplis, c'est le robot minimal dans une configuration permettant une dextérité maximale qui aura la meilleure évaluation.

4.2.5 Algorithme Evolutionnaire multi-chromosomique

La simulation utilise une représentation mixte, avec un codage binaire/réel, pour la constitution du génotype. Celui-ci comprend donc des informations sur la topologie et la configuration des solutions. Le codage des solutions, les opérateurs évolutionnaires ainsi que leurs paramètres associés sont décrits ci-après. Enfin les conditions d'arrêt de la recherche des solutions et l'algorithme général pour la simulation, regroupant toutes les descriptions effectuées, sont donnés.

Codage des paramètres topologiques et configurationnels

Le génotype résulte de l'association de deux chromosomes (Fig. 4.5). Le premier concerne la topologie des solutions, il regroupe les informations sur les segments constitutifs de la chaîne cinématique (orientation relative de la liaison, type, longueur de la barre). Les gènes de ce chromosome utilisent un codage binaire sur 8 bits dont la

Paramètre	Possibilités	Nombre de bits
Orientation relative	Id, R_x, R_y, R_z	2
Type de la liaison	R, P, F	2
Longueur de la barre	16	4

TAB. 4.2: Codage binaire des paramètres topologiques

répartition s'effectue de la manière suivante (tableau 4.2) :

- 2 bits fixent l'orientation relative du repère lié au segment par rapport à celui du segment précédent. Ils offrent 4 possibilités : pas de changement, une rotation d'axe \vec{x} , ou \vec{y} , ou \vec{z} .
- 2 bits précisent le type de liaison utilisé : rotoïde (R), prismatique (P) ou fixe (F).
- 4 bits donnent la longueur de la barre (entre 0 et 1 mètre), soit $1/16=0.0625$ m, $2/16=0.125$ m,...

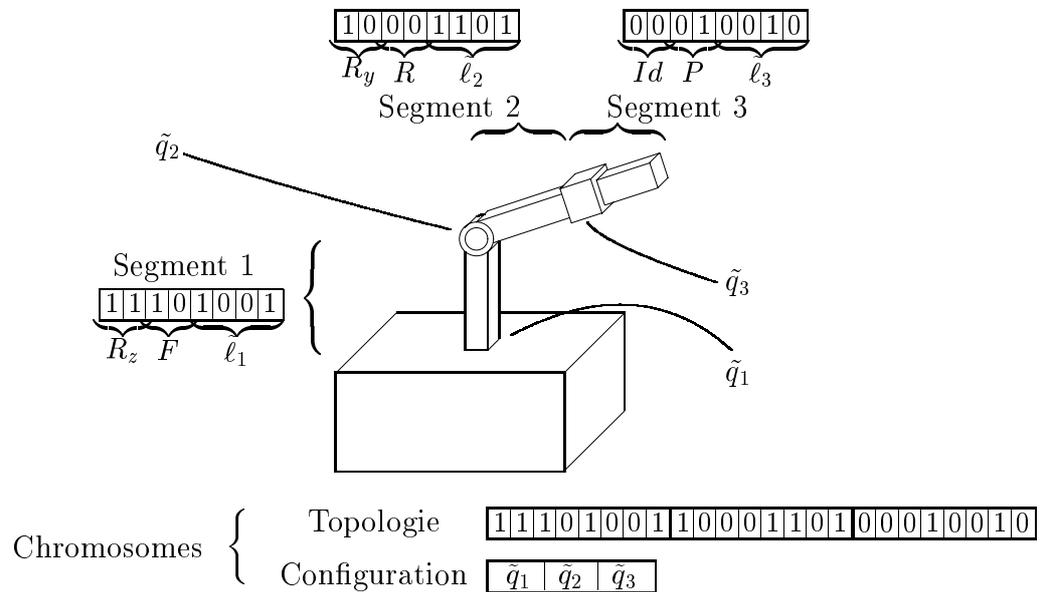


FIG. 4.5: Génotype mixte dans le cas d'un robot à 3 ddl.

Ce codage est non redondant dans le sens où chaque bit est nécessaire à la représentation d'un segment. Cependant l'évolution peut être légèrement gênée par son aspect dégénéré qui provient du fait que 2 bits (donc 4 solutions possibles) sont utilisés pour

le type de liaison (3 possibilités utilisées). L'augmentation correspondante de l'espace de recherche ne devrait pas modifier la convergence de l'algorithme. Le chromosome global lié à la topologie résulte de la concaténation de ces chaînes binaires (Fig. 4.5). Sa longueur est donc de $n_b * 8$ où n_b est le ddl de la structure cinématique.

Le codage des configurations utilise directement n_b valeurs réelles, \tilde{q}_i ($i = 1, \dots, n_b$), comprises entre 0 et 1. Elles correspondent aux valeurs articulaires q_i des liaisons, rapportées entre 0 et 2π . A chaque solution topologique sont donc associés n_t vecteurs réels de dimensions n_b , n_t représentant le nombre de sous-tâches à réaliser.

Opérateurs et paramètres évolutionnaires

Les opérateurs de mutation, croisement et sélection utilisés ont été testés avec succès sur une optimisation topologique orientée tâche en positionnement d'une chaîne cinématique à socle fixe. Ils assurent respectivement :

- une mutation bit-à-bit qui gère l'évolution de paramètres réels ou binaires en changeant la valeur d'un bit avec un paramètre de mutation p_m .
- un croisement uniforme macroscopique qui échange la valeur des bits entre deux parents avec une probabilité de croisement p_c .
- une sélection basée sur la méthode de la roulette et proportionnelle à la valeur de la fonction objective.

Bien que devant gérer deux évolutions séparées, les opérateurs utilisés sont les mêmes pour la manipulation des génotypes binaires et réels. Ces opérateurs se montrent performant du point de de la convergence de l'algorithme et des temps de calcul (un croisement à un point de césure sur les topologies, par exemple, augmente le temps de simulation de 10 % pour le même résultat).

La régénération se fait par remplacement de l'ancienne génération (les parents) par la nouvelle (les enfants). Les caractéristiques d'un individu modifié par les opérateurs évolutionnaires sont donc définitivement perdues (chap. 3). Si ce fait n'est pas gênant pour les topologies, les configurations doivent converger rapidement vers la solution, lorsqu'elle existe. L'opérateur de récupération présenté au chapitre précédent est donc

Paramètre	Notation	opt sup	opt inf
Taille de la population	k	50	50
Génération	g	50	30
Croisement	p_c	0.5	0.75
Mutation	p_m	0.01	0.12
Récupération	p_r	0	1

TAB. 4.3: paramètres d'évolutions pour la simulation

à nouveau utilisé dans l'optimisation inférieure pour réintroduire le meilleur individu,

toutes générations confondues, et le faire participer à la création des nouvelles populations configurationnelles. L'équilibre entre les rôles joués par les boucles d'optimisation inférieure et supérieure est géré par les paramètres évolutionnaires liés aux opérateurs correspondants. Ceux-ci sont résumés dans le tableau 4.3. Ils ont été définis de manière à assurer une convergence rapide de l'algorithme inférieure d'une part, d'autre part l'émergence fréquente de nouvelles solutions topologiques dans la boucle supérieure.

Conditions d'arrêt et d'élimination

Optimisation supérieure : Deux conditions d'arrêt sont utilisées pour la boucle d'optimisation supérieure. Elles ne sont pas particulièrement restrictives du fait de la recherche diversifiée des topologies. La première vérifie l'avancement générationnel et la boucle s'arrête lorsque le nombre maximal des générations est atteint. La seconde teste le nombre de ddl décroissants pour les solutions. L'initialisation de départ utilise 6 ddl pour les solutions (donc un chromosome de 6 octets) et ce nombre décroît de 1 à chaque fois que la condition d'arrêt 1 est vraie. La condition 2 est vraie lorsque le nombre de ddl passe à 0.

Optimisation inférieure : Pour la recherche des configurations, la condition d'arrêt devient vraie lorsque la fonction coût inférieure (F_{config}) atteint sa valeur maximale, à un coefficient de précision demandée près, ou lorsque le nombre de génération est à son maximum.

Élimination systématique Une condition sévère d'élimination est mise en place dès l'initialisation de la population topologique, et utilisée à chaque génération. Elle consiste à ne pas étudier ou reproduire des solutions qui ne pourront (quelle que soit la configuration) réaliser la tâche immédiate de positionnement, ou ultérieure, de déplacement autour de la position de l'effecteur. Les solutions concernées sont celles dont la longueur totale est inférieure à la distance minimale à l'objectif. Si une solution décodée ne satisfait pas cette condition, elle est immédiatement remplacée par une solution nouvelle, déterminée aléatoirement.

Algorithme évolutionnaire général

L'algorithme général correspondant à l'élimination des structures incompatibles, basée sur la synthèse préliminaire, est donné sur la figure 4.6. Il résume les différentes étapes détaillées précédemment. Trois parties principales se distinguent :

- l'optimisation supérieure, qui gère l'évolution des topologies ;
- l'optimisation inférieure, qui associe à chaque topologie une configuration adaptée, lorsqu'elle existe ;
- la sauvegarde des paramètres pour les solutions présentant un bon comportement.

Cette dernière est totalement basée sur une moyenne sur l'ensemble des sous-tâches de l'évaluation inférieure. L'établissement du seuil de tolérance F_α est discuté par la suite. La sauvegarde comprend un ensemble d'informations utiles pour la suite de l'étude

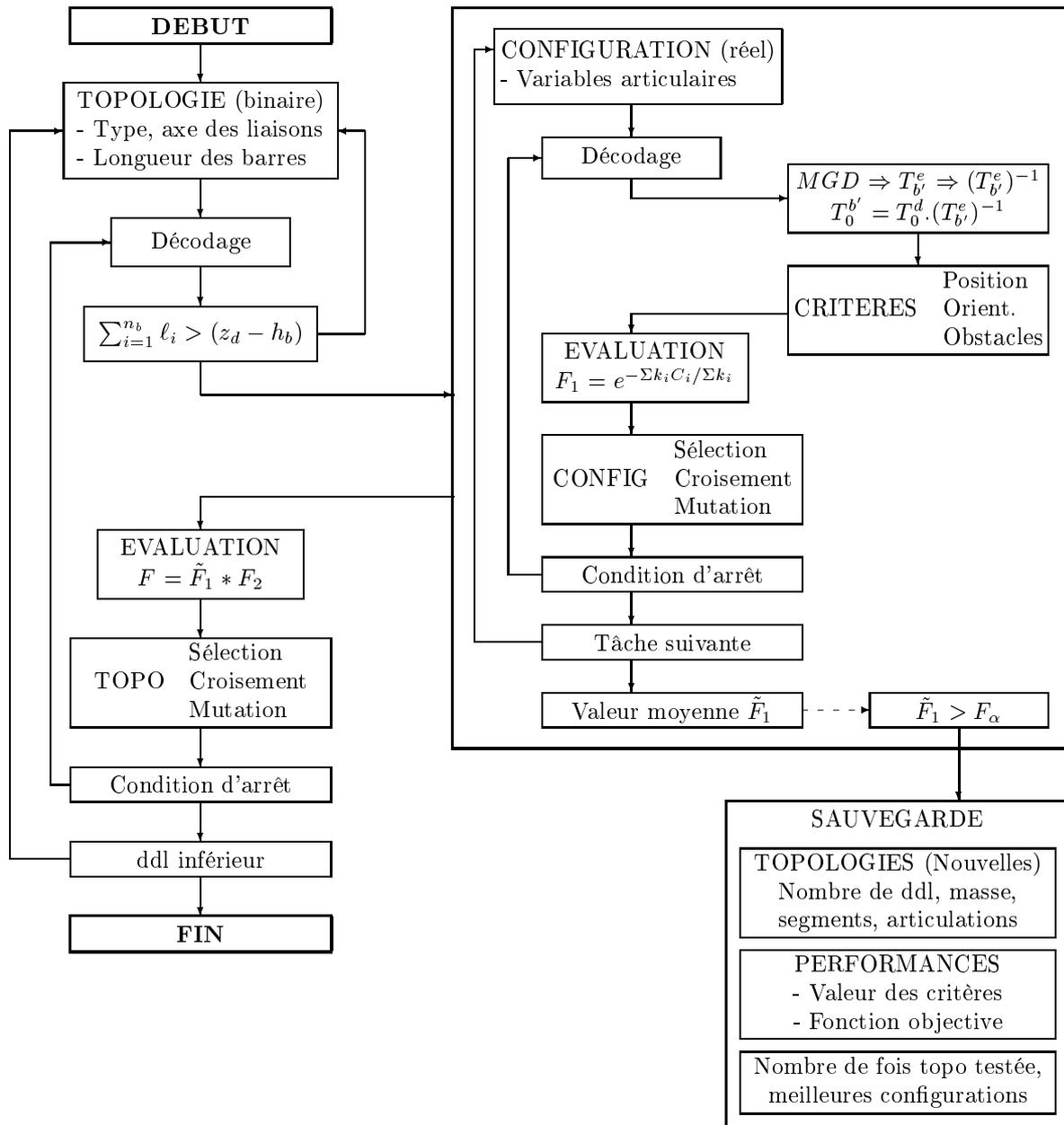


FIG. 4.6: Synthèse de mécanismes pour l'élimination préliminaire.

(optimisation finale des structures compatibles), mais aussi pour la compréhension des solutions proposées :

- Toute solution topologique sous sa forme codée, son ddl.
- La valeur de ses performances pour chaque évaluation : supérieure, inférieure, inférieure moyenne, globale, ainsi que la valeur des différents critères.
- Si la structure a déjà été testée par l'algorithme inférieur, son nombre d'apparition et les 5 meilleures configurations qui lui sont associées.

L'enregistrement des solutions topologiques ne prend effet que lorsque la structure cinématique n'apparaît pas dans la liste déjà existante, c'est-à-dire lorsque les 4 premiers bits de chaque gène du chromosome topologique diffèrent de ceux des individus déjà présents. Dans le cas positifs, leur performances sont comparées et le meilleur individu est conservé avec ses caractéristiques (longueur des barres, performances). La conservation des 5 meilleures configurations des individus déjà testés a pour but de réduire l'étude par la ré-injection de ces configurations dans l'évaluation inférieure. Ainsi, si la valeur optimale de l'adaptation inférieure a déjà été obtenue, la convergence est immédiate et l'individu suivant peut être testé. Dans le cas contraire, une chance supplémentaire permet d'obtenir l'adaptation maximale.

4.3 Résultats de simulation

4.3.1 Tâche et sous-tâches

La tâche générale est un suivi de trajectoire semblable à celui décrit au chapitre précédent : la trajectoire désirée est horizontale rectiligne le long d'un mur qui présente une marche verticale de 50 centimètres (figure 4.7). L'orientation de l'organe terminal est prise en compte et désirée perpendiculaire à la trajectoire de son début à sa fin. Deux cas sont étudiés : sans effort demandé à l'effecteur dans un premier temps, puis avec un effort de 2 newtons dans la direction de l'organe terminal vers le mur dans un second temps.

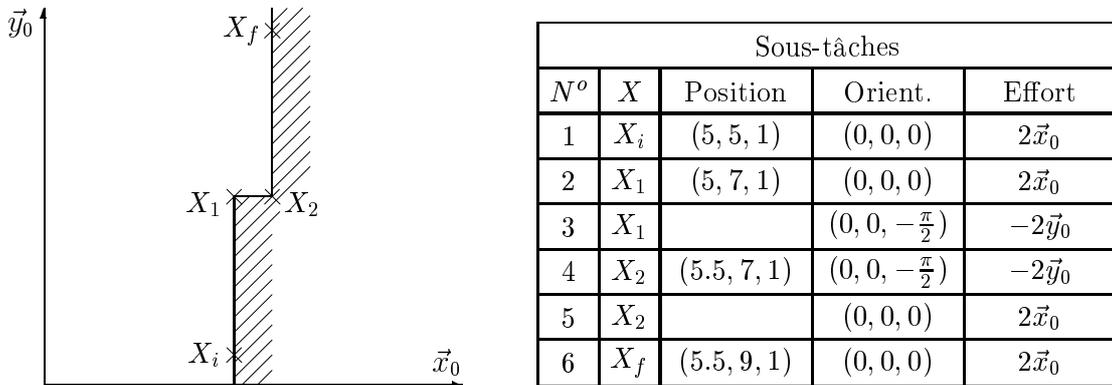


FIG. 4.7: La tâche et sa décomposition.

Les caractéristiques de la tâche peuvent donc être énoncées de la manière suivante, comprenant des contraintes liées à l'objectif et à l'environnement :

- Objectif : suivi d'une trajectoire par l'effecteur allant de $X_i = (5, 5, 1)$ à $X_f = (5.5, 9, 1)$ dans le référentiel fixe \mathcal{R}_0 . L'orientation de l'effecteur doit être perpen-

diculaire à la paroi. Lorsqu'il est demandé, un effort de $2N$ est à appliquer à l'effecteur, perpendiculaire à la paroi et orienté vers celle-ci.

- Environnement : outre le sol plat horizontal, il contient un obstacle représenté par un assemblage de surfaces verticales interdites. Le mur ainsi modélisé présente un décrochement vertical de 50 cm allant de X_1 à X_2 (Fig.4.7).

En considérant la décomposition de la tâche globale en sous-tâches caractéristiques introduite en début de chapitre, 6 sous-tâches distinctes ont été choisies pour la représentation. Elles sont résumées sur le tableau de la figure 4.7, et permettent d'éliminer toute structure qui ne pourra pas cinématiquement satisfaire les impératifs liés à l'objectif ou l'environnement.

La tâche décrite ici est une tâche simple qui regroupe une problématique en position, orientation et effort. Elle sert de tâche de base à notre recherche, de manière à valider les solutions proposées par le simulateur dans un cas simple. Elle subira des modifications qui permettront de mettre en évidence la pertinence des résultats recherchés (le respect des contraintes par exemple, ou un suivi de trajectoire avec variation des trois dimensions et angles...).

Sous-tâches				
N^o	X	Position	Orient.	Effort
1	X_i	(5, 5, 1)	(0, 0, 0)	$2\vec{x}_0$
2	X_1	(5, 7, 1)	$(0, 0, -\frac{\pi}{2})$	$-2\vec{y}_0$
3	X_2	(5.5, 7, 1)	$(0, 0, -\frac{\pi}{2})$	$-2\vec{y}_0$
4	X_f	(5.5, 9, 1)	(0, 0, 0)	$2\vec{x}_0$

TAB. 4.4: Les quatre sous-tâches caractéristiques restantes.

Le nombre de sous-tâches nécessaires augmente très rapidement avec la complexité de la tâche. Dans notre cas la décomposition s'avère simple : une sous-tâche est créée à chaque point de changement statique. On peut cependant la simplifier d'avantages en considérant l'équivalence complète entre les sous-tâches proposées, puisque la trajectoire désirée est une translation de l'organe terminal parallèlement au sol. Les sous-tâches 1 et 2 par exemple peuvent être vues comme identiques et l'étape n°2 peut être supprimée car les modifications de la géographie ne portent pas à conséquences dans ce cas. Il en va de même pour la sixième étape, qui peut être assimilée à la première. Le robot doit cependant se conformer aux contraintes d'environnement, et adopter une configuration qui lui permette d'éviter la collision avec le mur (notamment lors du franchissement de la marche). Les sous-tâches liées à notre étude se résument donc à celles reportées dans le tableau 4.4.

4.3.2 Comportement de l'algorithme

Hiérarchie des contraintes

Les différentes contraintes imposées aux individus ont été classées de manière à ce que l'évaluation d'une solution passe par plusieurs paliers. Ceux-ci fixent l'importance relative des critères dans l'évaluation :

$$\left. \begin{array}{l} 1) \text{ Situation} \\ 2) \text{ Obstacle} \\ 3) \text{ (Effort désiré)} \end{array} \right\} \tilde{F}_{\text{confg}} : \left. \begin{array}{l} \text{Performance} \\ 4) \text{ Mobilité} \\ 5) \text{ Structure min} \end{array} \right\} F : \begin{array}{l} \text{Performance,} \\ \text{confort et} \\ \text{minimalisme} \end{array}$$

L'intérêt de ces seuils est multiple. D'une part, ils empêchent que les critères s'équilibrent au sein de la fonction d'adaptation, et que l'un soit favorisé au détriment de n'importe quel autre. Pour exemples, la mobilité d'une solution peut l'entraîner sur la précision du positionnement, ou un contact aux obstacles peut être considéré de la même manière qu'une structure lourde. L'établissement de cette hiérarchie au sein des évaluations successives est donc nécessaire. Elle permet de différencier efficacement et rapidement les contraintes impératives (liées à la réalisation de la tâche) des contraintes qui doivent s'équilibrer. Par ordre d'importance croissante, leur classement est le suivant :

1. Réalisation de la tâche ($k_L, k_{A_1}, k_{A_2}, k_O$)
2. Confort dans la réalisation (k_f)
3. Minimalisme structurel (k_P)

où les k_i sont les coûts liés à chaque critère dans la fonction d'adaptation. Les valeurs qui leur ont été attribuées pour la simulation sont reportées dans le tableau 4.5, et mettent en évidence les remarques précédentes. Notons que le caractère binaire du

Critère	Symbole	Coût
Altitude	k_L	2
Rotation	k_{A_1}, k_{A_2}	2
Obstacle	k_O	1
Poids	k_P	0.5
Mobilité	k_f	0.5

TAB. 4.5: Coût lié à chaque critère.

critère d'évitement d'obstacles permet lui autorise un gain relatif faible sans lui retirer de son efficacité. Le critère de poids est quant à lui considéré comme d'importance relative moindre du fait de la révision ultérieure des variables de conception. Le critère lié à la mobilité fait l'objet d'un traitement particulier. Celui-ci intervient en effet sur les deux niveaux dans l'évolution des solutions :

- déterminer une meilleure configuration à une solution topologique donnée (optimisation inférieure);
- trouver un compromis entre le caractère minimaliste d'une structure et son confort dans l'atteinte d'une situation (optimisation supérieure).

La valeur du gain associé doit donc être égale à celle liée au poids sans empiéter sur la performance de la tâche de manière à ce qu'un équilibre puisse exister.

Convergences

Les convergences des boucles d'optimisation supérieure et inférieure de l'algorithme à deux niveaux sont montrées sur les figures 4.8(a) et 4.8(b) respectivement. Elles concernent la recherche de solutions à 6 ddl sur les 4 sous-tâches décrites précédemment. L'évolution supérieure (figure 4.8(a)) utilisant la fonction d'évaluation globale F montre

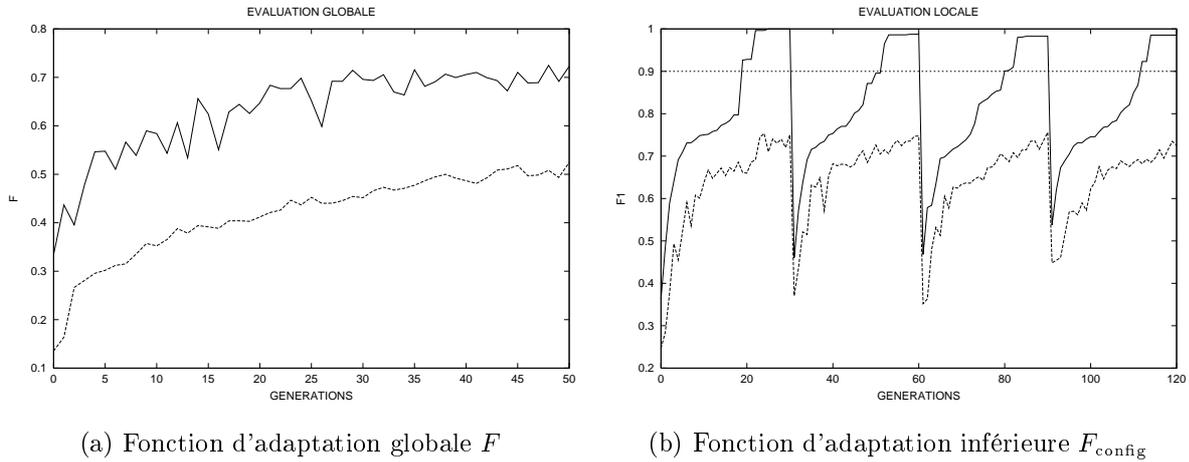
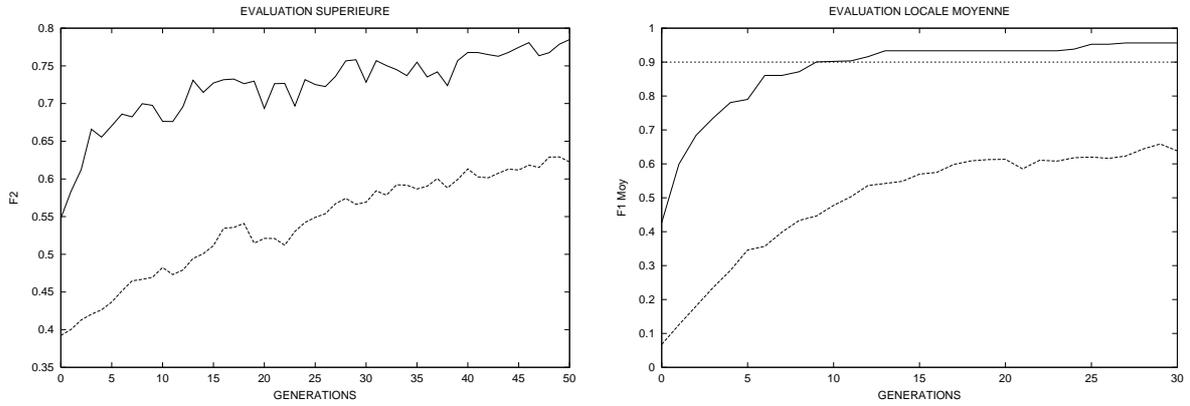
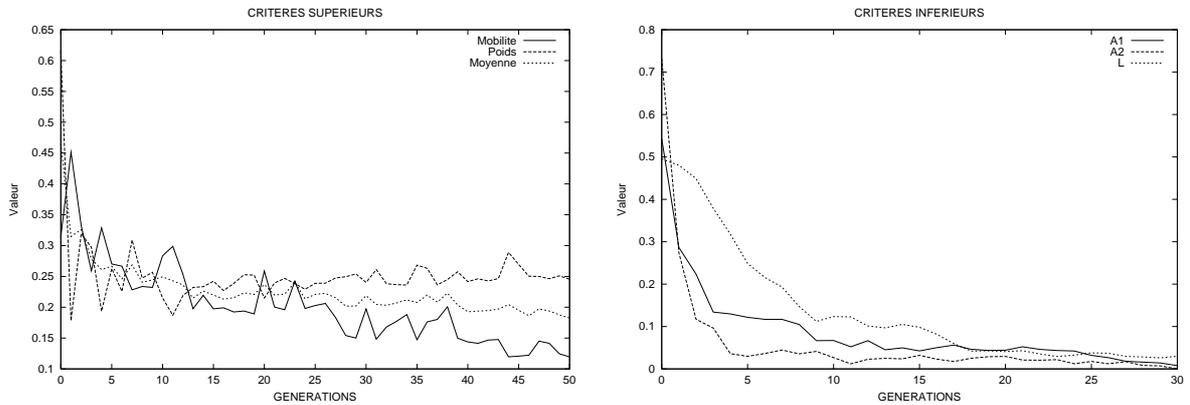


FIG. 4.8: Adaptation globale et locale pour la synthèse par optimisations imbriquées.

une augmentation rapide des valeurs d'adaptation du meilleur individu et de la moyenne des individus jusqu'à la vingtième génération car les valeurs de départ sont relativement basse. La croissance est ensuite lente jusqu'à la cinquantième génération. Les valeurs d'adaptation finales se situent au-dessus de 0.8 pour le meilleur individu et de 0.6 pour la moyenne de la population.

L'évolution inférieure liée aux configurations montre une convergence rapide, lorsqu'une solution existe, en accord avec la dynamique forte provoquée par les paramètres évolutionnaires inférieurs (figure 4.8(b)). L'évolution correspondante de la valeur moyenne $\tilde{F}_{\text{config}}$ est représentée sur la figure 4.9(b) en fin d'évolution des topologies (ce qui explique la très bonne valeur d'adaptation de la moyenne de la population).

L'opposition rigidité/souplesse dans les évaluations inférieure/supérieure apparaît particulièrement sur l'évolution de la valeur des critères intervenant dans chacune des

(a) Evaluation supérieure F_2 (b) Evaluation moyenne inférieure $\tilde{F}_{\text{config}}$ 

(c) Evolution des critères de la fonction supérieure (mobilité et poids)

(d) Evolution des critères de la fonction inférieure (situation et obstacles)

FIG. 4.9: Evaluations intermédiaires, critères inférieurs et globaux

fonctions d'adaptation (figure 4.9). En effet, si tous les critères inférieurs tendent vers 0 simultanément, ceux de la boucle supérieure trouvent un équilibre entre leur deux valeurs. Nous nous intéressons essentiellement au comportement de l'algorithme inférieur, puisque c'est au cours de cette optimisation que la faisabilité de l'ensemble des sous-tâches par la structure étudiée est vérifiée ainsi que la distinction qualitative de l'association topologie/configuration. Les conditions de sauvegarde, ainsi que les erreurs comises sur la situation de l'effecteur sont discutées ci-après. Le temps de calcul pour une simulation complète, pour les 50 générations de l'optimisation supérieure sur 50 individus, incluant chacune 50 individus * 30 générations sur 50 individus pour l'optimisation inférieure, est de 44 minutes sur station Sun Ultra 5. Ce temps est particulièrement faible du fait des conditions d'arrêt et de la dynamique évolutionnaire fixés en conséquence.

4.3.3 Solutions topologiques retenues

Elimination et sélection

Deux conditions sont utilisées pour la non sélection des solutions topologiques : La première intervient dès l'étape d'initialisation (voir l'algorithme général, figure 4.6). Elle effectue l'élimination et le remplacement systématiques des solutions présentant des chaînes cinématiques trop courtes.

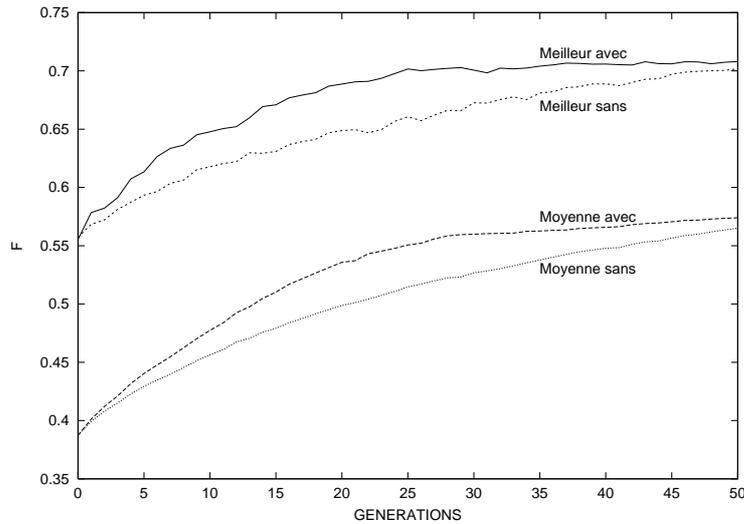


FIG. 4.10: Statistiques de convergence, sur 100 simulations, avec et sans la contrainte initiale de remplacement des topologies.

Entre 2 et 15 % de ces cas apparaissent dès l'initialisation, en fonction du nombre de ddl des robots considérés, et 2 à 5 % à chaque nouvelle génération. Ces solutions sont immédiatement remplacées par des individus initialisés aléatoirement, introduisant des nouveaux génotypes dans l'évolution. La régénération de la population est donc encouragée par cette intervention. Les graphes de la figure 4.10 comparent les évolutions de l'adaptation supérieure avec et sans cette condition. Si la valeur de convergence est identique dans les deux cas, la vitesse de convergence est améliorée. Autre conséquence positive, la quantité de solutions différentes retenues augmente de 26 % en moyenne sur des statistiques effectuées sur 100 simulations (tableau 4.6).

La deuxième condition de sélection se base sur l'évaluation des configurations des individus et agit lors de l'optimisation inférieure. Cette condition utilise la valeur moyenne $\tilde{F}_{\text{config}}$ sur toutes les sous-tâches de la fonction de performance F_{config} . Une solution sera conservée (donc sauvegardée pour l'étude ultérieure) si sa valeur moyenne associée est supérieure ou égale au seuil de tolérance à l'erreur F_{α} . Celui-ci est calculé sur la base d'une solution qui parvient à l'objectif mais entre en contact avec l'environnement :

$$F_{\alpha} > F_{\text{config}}(L = A_1 = A_2 = 0, O = 1) = 0.86$$

$$\text{Elimination : } \tilde{F}_{\text{config}} < F_{\alpha} = 0.90$$

Les solutions non sauvegardées à l'issue de la boucle d'optimisation inférieure sont ré-introduites dans le cycle des générations, mais non enregistrées comme solutions robotiques potentielles. Le tableau 4.6 donne les minima et maxima de topologies différentes sauvegardées sur 100 simulations effectuées. La majorité des propositions reste à 6 ddl. L'abondance des propositions à 5 ddl montre que le degré de liberté

		Régénération initiale		
		Sans	Avec	Augmentation
ddl	1	0	0	-
	2	0	0	-
	3	0/ 3 /5	2/ 7 /9	+ 133 %
	4	4/ 10 /22	12/ 22 /30	+ 120 %
	5	18/ 42 /50	30/ 48 /55	+ 14 %
	6	32/ 49 /59	40/ 54 /65	+ 10 %
Total		54/ 104 /136	84/ 131 /159	+ 26 %

TAB. 4.6: Quantité minimale/**moyenne**/maximale des solutions sauvegardées par type de structure, avec et sans la contrainte initiale (statistiques sur 100 simulations).

supplémentaire introduit par la rotation autour de \vec{z} est pris en compte et exploité. Les robots à 4 ddl proposés correspondent à des cas spéciaux de positionnement de la plate-forme, ainsi que ceux à 3 ddl (beaucoup plus rares). Les manipulateurs à 1 ou 2 ddl sont totalement inexistantes. Ce sont donc les chaînes cinématiques les plus complexes qui ont plus de facilités à se faire sélectionner, ce qui était prévisible puisque la situation à atteindre en X_2 (figure 4.4) demande une certaine dextérité pour éviter la collision avec le mur.

Précision

Les cinq critères sur lesquels sont basées les sauvegardes des topologies sont correctement exploités et les erreurs sur les contraintes géométriques sont très faibles à la fin de l'optimisation inférieure. Le tableau 4.7 dresse l'inventaire des performances de la fonction objective inférieure et de sa moyenne ainsi que des erreurs par rapport aux paramètres de situation et d'effort. Les erreurs sur θ et ψ sont nulles dans 60% des cas et ne dépassent pas $10^{-3}\pi$ radians. La figure 4.11 montre la cote pour tous les manipulateurs sélectionnés au cours d'une simulation (toutes topologies confondues, chaînes cinématiques à 6 ddl). On y observe que l'erreur maximale commise n'excède pas le millimètre ce qui est convenable. La précision reste donc très grande. Le seuil de tolérance F_{α} permet cependant de donner une chance à des structures dont les variables de conception sont portées à révision.

	Statistiques		Meilleur		Moins bon	
	Meilleur	Moyenne	Meilleur	Moyenne	Meilleur	Moyenne
\tilde{F}_1	0.976	0.696	0.991	0.769	0.925	0.624
F_1	0.976	0.696	1	0.782	0.896	0.777
ε_z (mètres)	0.0008	0.006	0	0.005	0.0011	0.011
ε_θ (degrés)	0.110	1.652	0	0.086	1.268	2.256
ε_ψ (degrés)	0.110	1.652	0	0.035	0.854	3.659

TAB. 4.7: Performances et erreurs produites pour le meilleur individu et pour la moyenne de la population, par rapport à la tâche en position, orientation, effort pour les statistiques sur 100 simulations, la meilleure et la moins bonne simulation.

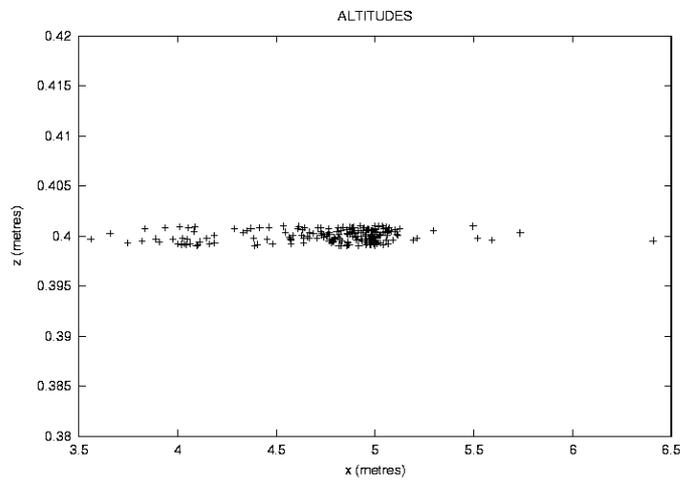


FIG. 4.11: Cotes des extrémités inférieures des chaînes cinématiques.

4.4 Conclusion

L'objectif de l'élimination préliminaire est de supprimer un ensemble de solutions robotiques inadaptées à la réalisation d'une tâche donnée, ce afin de limiter l'étude ultérieure aux seules solutions pouvant potentiellement réaliser la tâche. Le travail réalisé dans ce chapitre répond à ces exigences, puisqu'il permet :

- dans le cadre d'une étude minimale,
- de trouver une grande diversité de structures
- qui réalisent des parties représentatives de la tâche
- avec une erreur faible.

Pour obtenir ces résultats, une méthodologie a été développée, dont les caractéristiques principales sont les suivantes :

- La décomposition de la tâche globale en sous-parties qui la caractérisent a permis :

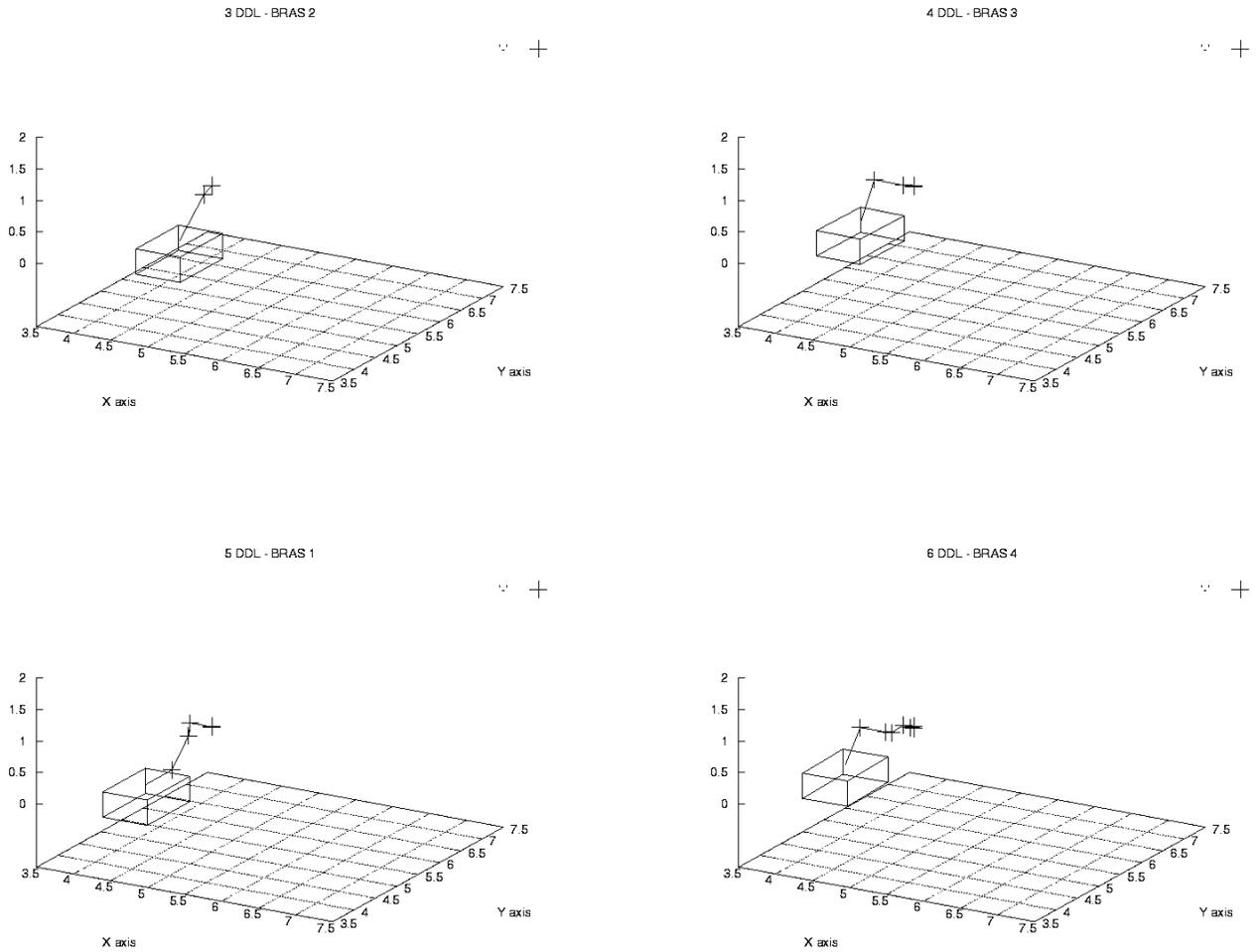


FIG. 4.12: Exemples de manipulateurs proposés.

- de limiter l'étude au cadre statique ;
- de s'inspirer du travail effectué sur la synthèse classique de chaînes cinématiques, en l'occurrence
- l'utilisation de structures modulaires permettant une représentation mathématique simple par transformations homogènes.
- Un algorithme évolutionnaire à deux niveaux a été développé, avec :
 - une optimisation supérieure liée à une évaluation souple des topologies donnant accès à un équilibre mobilité maximale à l'effecteur / poids du robot minimal. Ce niveau permet l'émergence de nombreuses structures différentes.
 - une optimisation inférieure dotée d'une évaluation rigoureuse sur les contraintes de sous-tâches afin de déterminer, si elles existent, les solutions configurationnelles associées.

Du fait de l'agencement aléatoire des modules, l'espace de recherche des solutions topologiques est entièrement recouvert. Les structures retenues sont donc de tout ordre (figure 4.12). Celles-ci sont évaluées avec leurs configurations par un ensemble de critères locaux uniquement liés à la sous-tâche : position, orientation, obstacles. Leur sélection entraîne la sauvegarde de toutes leurs caractéristiques (topologiques, mathématiques, configurationnelles) et l'ensemble des solutions est ré-introduit dans le cycle évolutif. Une contrainte supplémentaire d'élimination stricte, basée sur l'incapacité d'une structure à répondre aux exigences des sous-tâches quelle que soit sa configuration, et de remplacement de cette solution par l'initialisation d'une autre, permet de booster la régénération des générations et d'obtenir d'avantages de solutions différentes.

L'algorithme en deux étages proposé et décrit ci-avant se base donc essentiellement sur une évaluation orientée sur l'objectif du robot. Il n'est pas réellement fait mention d'une contrainte forte ("dure") permettant la préférence d'une configuration sur une autre. Aucune condition n'a été posée, par exemple, sur la position ou l'orientation de la plate-forme. Cette liberté est principalement destinée à l'émergence et à la sauvegarde de nombreuses solutions, ou plus précisément à l'élimination stricte des solutions topologiques incompatibles à la réalisation de la tâche. Parallèlement, il n'existe pas vraiment une seule solution configurationnelle pour des structures redondantes du type manipulateurs mobiles, sauf dans les cas où les contraintes de la tâche, ou fixées par l'utilisateur, sont trop fortes pour laisser le choix au robot. L'approche est adaptée à notre problème de pré-élimination des structures incompatibles à la réalisation de la tâche : les évolutions des fonctions d'adaptation supérieure et inférieure se complètent de manière à fournir

- une convergence inférieure rapide vers une situation exacte validée par une évaluation locale ;
- une convergence supérieure plus lente et proposant un équilibre entre la structure minimale et la mobilité maximale pour les solutions cinématiques ainsi que de nouvelles possibilités.

Les différents critères utilisés permettent la sélection de chaînes cinématiques qui répondent aux exigences *statiques* de la tâche (positionnement, effort) mais restreignent suffisamment peu la recherche pour que cet ensemble de solutions reste très varié. Certaines chaînes cinématiques de la sélection ne permettront pas la réalisation convenable de la tâche car trop spécifiques au positionnement particulier demandé. En effet le fait de pouvoir atteindre deux positions et orientations ne garantit pas la possibilité de suivre la trajectoire entre ces deux situations.

L'analyse, effectuée en statique, doit être complétée de la commande des actionneurs du manipulateur mobile de manière à contrôler complètement la réalisation de la tâche et vérifier la gestion de la redondance du système. On peut dire, pour le moment, que l'ensemble de structures retenues ont passé une étape de pré-sélection en vue de la tâche de manipulation mobile. Ces structures seront complètement testées lors de la sélection finale dans le chapitre suivant.

Chapitre 5

Optimisation commande/structure

5.1 Problématique générale

5.1.1 Avancement / Objectif

L'étape précédente a consisté à éliminer toutes les structures de types chaînes cinématiques qui présentaient des incompatibilités avec la réalisation de la tâche. Ce chapitre est destiné à la sélection finale d'une solution. Le modèle complet, plate-forme mobile et chaîne cinématique, est utilisé. L'approche est toujours orientée vers la réalisation de la tâche. Elle consiste donc à effectuer une analyse de la faisabilité globale en testant les structures précédentes et la plate-forme mobile sur le suivi global (figure 5.1). Les variables recherchées sont les paramètres de consignes. Cette première

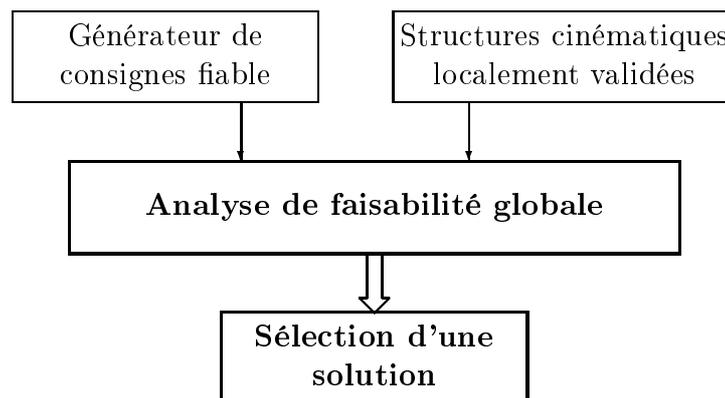


FIG. 5.1: Avancement de l'étude.

partie du chapitre analyse le problème actuel en vérifiant la compatibilité de l'étude avec la précédente. Un bilan des variables, connues en entrée et désirées en sortie, est tout d'abord dressé, de manière à établir les extensions et les limites de l'étude. Le

problème particulier de la configuration initiale des robots pour effectuer le suivi est ensuite traité.

5.1.2 Bilan des données

Les robots utilisés pour la recherche sont constitués de la plate-forme mobile non-holonyme RoMAin et d'une des chaînes cinématiques précédemment sélectionnées. La tâche sur laquelle ces structures seront testées est celle du suivi de trajectoire avec franchissement de marche décrite au chapitre 3.

Les informations issues de l'étape d'élimination précédente se divisent en variables structurelles et configurationnelles. Elles constituent l'ensemble des variables entrée de la recherche actuelle :

Données en entrée

1. **Structure** : ddl, paramètres DH, $(a_i, \alpha_i, d_i, \theta_i)$, pour chaque liaison ($i = 1..n$), donc $n_b \times 3 + 1$ paramètres, en considérant l'une des variables comme liée au mouvement (θ_i pour une liaison rotoïde ou d_i pour une prismatique, annexe 2).
2. **Configuration** : *une* configuration, pour chaque sous-tâche, qui vérifie les équations de contraintes de la structure globale, mais que l'on peut caractériser d'arbitraire.

Les variables désirées en sortie concernent la description complète de la tâche robotique. Outre les informations structurelles précédentes, elles fournissent les informations liées à la réalisation de la tâche :

Données en sortie

1. **Structure** : ddl, paramètres DH pour chaque liaison, donc $n_b \times 3 + 1$ variables de structure.
2. **Consignes** : $n = 1..10$ vecteurs de consignes de dimension n_b , donc $n \times n_b + 1$ paramètres de commande.
3. **Configuration initiale** :
 - position et orientation de la plate-forme mobile (3 paramètres)
 - configuration de la chaîne cinématique (n_b paramètres)
 donc $3 + n_b$ paramètres de configuration initiale.

Les variables liées à la structure sont déjà déterminées (chaînes cinématiques de topologie établie). L'algorithme décrit au chapitre 3 s'est montré performant pour déterminer les inconnues liées aux déplacements d'un robot particulier. Deux problèmes restent donc à résoudre pour achever la sélection : dans un premier temps, vérifier l'adaptabilité de l'algorithme précédent au problème actuel. Dans un second temps, une attention particulière est à porter sur l'ensemble des variables qui déterminent la configuration initiale optimale du point de vue du déplacement.

5.2 Recherche des consignes : stabilité en complexité

Le problème de la recherche des entrées de commande est élargi au cas du suivi en position et orientation par l'effecteur du manipulateur. Son énoncé et les conditions de réalisation sont les mêmes que précédemment (chapitres 2 et 3) : nous allons ici vérifier la stabilité de l'algorithme face à un changement de dimension des chromosomes manipulés. L'étude peut donc se limiter à la recherche d'un jeu de consignes sans l'étendre au cas de la manœuvre puisque, dans ce dernier cas, aucune données supplémentaires se sont extraites du point de vue de la réalisation de la tâche. Nous avons en effet montré au chapitre 3 que si le suivi local de trajectoire est assuré, la recherche de consignes pour la manœuvre ne pose pas de problème.

Les rotations du poignet sont libérées une à une jusqu'à la mobilité maximale du bras manipulateur (6). La taille des chromosomes passe donc de 5 à 8 gènes, et la dimension de l'espace de recherche passe de \mathbf{R}^5 à \mathbf{R}^8 . Les problèmes d'optimisation que nous devons résoudre s'énoncent de la même manière que précédemment :

$$\begin{aligned} \text{Trouver } & \mathbf{U}^* = U(\mathbf{q}^*) \in \mathbf{R}^5 \\ \text{tel que } & F(\mathbf{X}_e^*) \geq F(\mathbf{X}_e) \quad \forall X_e \in \mathbf{R}^3 \end{aligned}$$

avec \mathbf{q} Vecteur des coordonnées généralisées
 \mathbf{X}_e Vecteur des coordonnées opérationnelles
 U Vecteur des consignes sur les actionneurs
 $F(\mathbf{X}_e)$ Fonction d'évaluation des entrées de commande

Les espaces de recherche des solutions changent, pour les quatre cas que nous considérons (voir annexe 3 pour les notations) :

Problème 1 $U(\mathbf{q}) \in \mathbf{R}^5$ $X_e \in \mathbf{R}^3$ Suivi en position (x_e, y_e, z_e)
Problème 2 $U(\mathbf{q}) \in \mathbf{R}^6$ $X_e \in \mathbf{R}^4$ Suivi en position et orientation (ψ_e)
Problème 3 $U(\mathbf{q}) \in \mathbf{R}^7$ $X_e \in \mathbf{R}^5$ Suivi en position et orientation (ψ_e, θ_e)
Problème 4 $U(\mathbf{q}) \in \mathbf{R}^8$ $X_e \in \mathbf{R}^6$ Suivi en position et orientation (ψ_e, θ_e, ϕ_e)

Le tableau 5.1 résume les adaptations des paramètres évolutifs correspondant aux libérations successives des rotations. Les convergences sur 150 générations peuvent être comparées sur la figure 5.2. La courbe 5.2(a) compare les évolutions des meilleurs individus de la population dans les cas recherche des entrées de commande pour un suivi en position seul et pour un suivi en position / orientation. La courbe 5.2(b) concerne les valeurs d'adaptation moyennes pour tous les individus des deux populations. Les résultats de convergence sont similaires et montrent la stabilité de l'algorithme face à la complexité du problème. Le changement se ressent au niveau de l'ajout de la dimension de recherche (rotation), mais pas au niveau de la taille des chromosomes. La figure 5.2(a) montre en effet la même allure d'évolution pour tous les problèmes, avec un décalage de convergence entre le problème 1 et les problèmes 2, 3 et 4. Ces derniers, en revanche, présentent entre eux une similitude de convergences quasi-exacte. Le même délai supplémentaire de convergence, entre le problème 1 et les autres, peut être observé pour les évolutions des moyennes des valeurs d'adaptation des différentes populations

Problème	1	2	3	4
k	100			
g	120	150		
p_c	1			
p_m	0.11	0.13	0.14	
p_r	5			

TAB. 5.1: Valeurs des paramètres évolutifs en fonction du degré de liberté de la structure

(figure 5.2(b)). Nous retrouvons un comportement "classique" de convergence sur un temps plus grand pour les problèmes 2, 3 et 4 que pour le problème 1, c'est-à-dire un retard, par rapport à ce dernier cas, avant le début de la décroissance de la valeur moyenne.

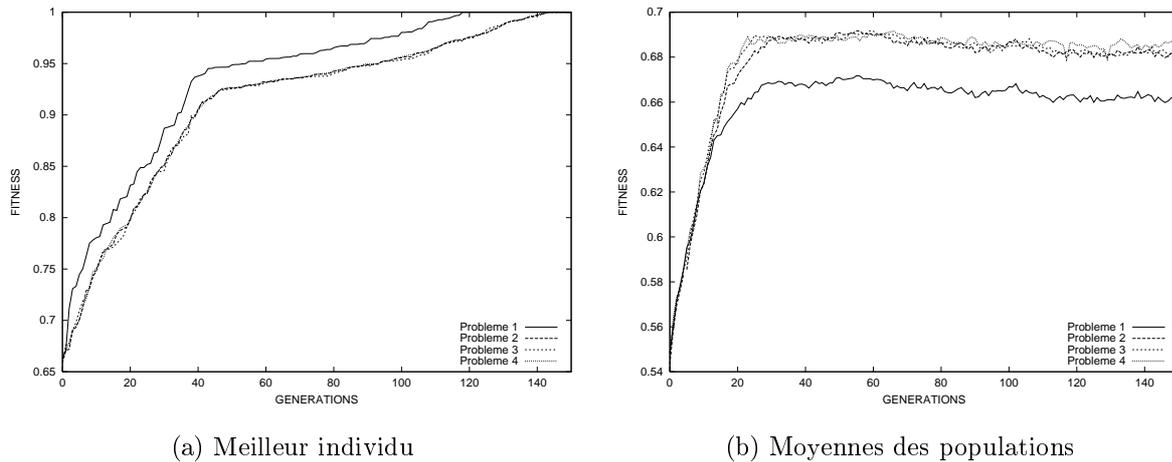


FIG. 5.2: Evaluations intermédiaires, critères inférieurs et globaux

Au cours de toute l'étude sur la recherche des entrées de commande, nous avons fait totalement abstraction du robot étudié. Bien sûr, des modèles dynamiques particuliers ont été utilisés, mais uniquement comme fonctions intermédiaires entre les entrées (vecteurs de consignes) et les sorties (valeurs des critères) d'un module d'évaluation. Pour l'optimisation en elle-même, seul le nombre de degrés de libertés de la structure importe réellement, afin de fixer la taille des chromosomes à l'initialisation. Le système étudié en lui-même n'importe pas. On peut donc affirmer que l'algorithme est stable dans sa recherche, fiable quant aux solutions proposées, et qu'il confère un caractère générique à l'approche. L'étude peut donc être étendue à n'importe quelle structure présentant les mêmes caractéristiques générales (chaîne cinématique ouverte sur une plate-forme mobile non-holonome à roues). Dans le même esprit, nous allons déterminer

une configuration initiale pour cette même structure et dans les mêmes conditions de réalisation de la tâche.

5.3 Détermination d'une configuration initiale

Ce problème n'a pas encore été abordé au cours de l'étude. En effet, l'établissement du générateur de consignes (chapitres 2 et 3) utilisait une configuration initiale arbitraire, posée de manière à ce que la plate-forme mobile soit dès le départ dans la direction désirée. L'élimination préliminaire (chapitre 4) génère quant à elle une configuration initiale aléatoire, s'assurant uniquement de la faisabilité locale (respect des contraintes géométriques) de la tâche.

5.3.1 problématiques de la recherche

Le problème qui se pose est sous l'influence de deux contraintes :

- la non-intervention, toujours, de l'utilisateur dans la recherche de l'algorithme interdit des simplifications arbitraires (du type position/orientation de la plate-forme données) qui résulteraient d'une connaissance *a priori* du type de solution attendue.
- la valorisation de la configuration initiale est fortement couplée à celle du suivi. On ne peut pas quantifier la qualité du vecteur opérationnel initial indépendamment de la génération de consignes du fait de l'évaluation globale du suivi de la trajectoire.

Nous considérons dans un premier temps la tâche de suivi en position sans manœuvre décrite au chapitre 2 et les mêmes robots sont utilisés. Les vecteurs impliqués sont les suivants :

Configuration	:	$(x_b, y_b, \theta, \phi_1, \phi_2, \phi_3)$
Coordonnées généralisées	:	$(\phi_d, \phi_g, \phi_1, \phi_2, \phi_3)$
Coordonnées opérationnelles	:	$(x_e, y_e, z_e, \psi_e, \theta_e, \phi_e)$

Le problème d'optimisation que nous devons résoudre peut s'énoncer de la manière suivante :

$$\begin{aligned} \text{Trouver} \quad & q_0^* = (\phi_1, \phi_2, \phi_3)^t \in \mathbf{R}^3 \\ \text{Tel que} \quad & f(U_1(q_0^b, q_0^*)) \geq f(U_1(q_0^b, q_0)) \quad \forall q_0 \in \mathbf{R}^3 \end{aligned}$$

q_0 représente alors le vecteur des coordonnées généralisées du bras manipulateur, q_0^b celui de la plate-forme mobile résultant. La configuration recherchée est un vecteur de dimension $(3 + n_b)$, qui assure :

- le respect des équations de contraintes de la structure permettant la correspondance entre la situation (position et orientation) de l'organe terminal et la situation initiale désirée ;
- le respect des contraintes liées à l'environnement.
- un positionnement favorable à la direction de déplacement désirée de manière à éviter une manœuvre inutile.

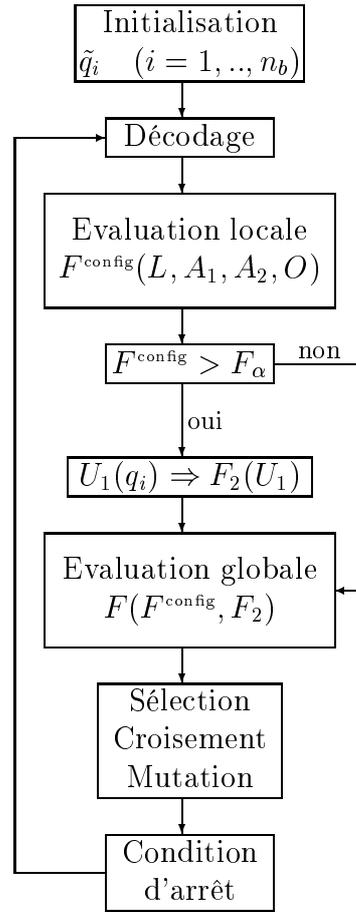


FIG. 5.3: Algorithme de recherche d'une configuration initiale permettant une valeur optimale du premier vecteur de consignes.

5.3.2 Algorithme

L'algorithme correspondant à cette recherche est donné sur la figure 5.3. Il est globalement similaire à celui utilisé pour la boucle d'optimisation inférieure de l'élimination préliminaire présentée au chapitre précédent et manipule les mêmes variables (vecteur des coordonnées généralisées de la chaîne cinématique). La configuration de la plate-forme (x_b, y_b, θ) est déduite de la même manière que précédemment de celle du manipulateur. La différence réside dans l'inclusion du vecteur d'entrées de commande.

Une sélection partielle est alors effectuée sur le premier vecteur de consignes :

$$U_1 = (u_1^1, \dots, u_1^{n_b+2})^t$$

pour tous les individus. Elle consiste à effectuer une simulation dégradée du générateur de consignes à ce seul premier vecteur, afin de tester celles-ci sur un temps très bref (10 incréments). Pendant ce laps de temps, la configuration qui résout le problème géométrique inverse et donne la meilleure évaluation au vecteur d'entrées de commande

est déterminée. Il est cependant important que cette recherche n'influe pas sur les consignes ultérieures et garde un caractère indépendant.

Une première évaluation vérifie la validation des contraintes de structure et d'environnement (figure 5.4). Elle utilise la fonction d'adaptation inférieure F^{config} explicitée

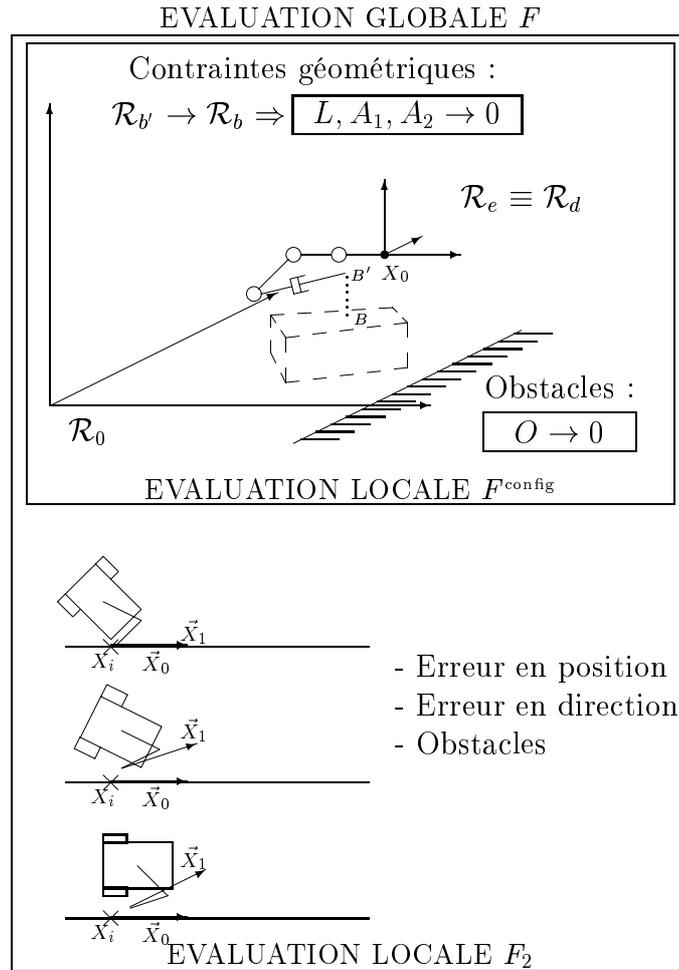


FIG. 5.4: Les deux évaluations locales pour quantifier la valeur d'une configuration initiale.

page 104. L'obtention d'une valeur supérieure au seuil de tolérance F_α autorise le test des entrées de commande sur la structure complète véhicule/chaîne cinématique pendant 10 pas de temps. Ce test permet une deuxième évaluation qui utilise la fonction d'adaptation F_2 (figure 5.3). On ne peut pas utiliser ici la fonction d'adaptation mise en place au chapitre 3, qui suppose une trajectoire totalement définie (comprenant un point de départ et un point d'arrivée). L'évaluation que nous utilisons est donc basée sur la correspondance entre direction résultante de l'effecteur et la direction désirée.

Le vecteur unitaire associé à cette dernière, noté $\delta\vec{x}_d$, est connu (Fig. 3.10) :

$$\delta\vec{x}_d = \frac{\vec{X}_1 - \vec{X}_0}{\|\vec{X}_1 - \vec{X}_0\|}$$

Le vecteur unitaire donnant la direction prise par l'organe terminal s'obtient en considérant la situation atteinte au bout de 10 pas de temps :

$$\delta\vec{x} = \frac{\vec{X}_{10} - \vec{X}_0}{\|\vec{X}_{10} - \vec{X}_0\|}$$

et l'angle α entre ces deux vecteur s'obtient par leur produit scalaire, sans ambiguïté de signe quant à la direction prise dans le déplacement :

$$\alpha = \cos^{-1}(\delta\vec{x} \cdot \delta\vec{x}_d)$$

Le critère est donc directement défini par la valeur de α qui doit tendre vers 1 :

$$C'_1 = 1 - \frac{\alpha}{2\pi} \rightarrow 0$$

Il remplace le critère précédent d'avancement sur la trajectoire C_1 , et est associé aux deux autres critères C'_2 et C_3 (erreur en position/orientation et évitement de collision respectivement) dans la fonction $F^{\text{consignes}}$ présentée page 2.4.2. L'évaluation globale d'une configuration initiale correspond à la moyenne des deux évaluations précédentes :

$$F = \frac{F^{\text{config}} + F_2}{2}$$

avec $F_2 = 0$ dans le cas d'une solution vérifiant $F^{\text{config}} < F_\alpha$. Ainsi les solutions ne passant pas la première épreuve ont une chance d'être conservées dans le pool des géniteurs et éventuellement la possibilité de s'améliorer la génération suivante.

De nombreuses possibilités configurationnelles de départ sont obtenues, comme le montre sur l'exemple d'une simulation la figure 5.5. Celle-ci indique, pour l'ensemble des différentes configurations "valables" ($F^{\text{config}} > F_\alpha$) de la chaîne cinématique, les configurations (x_B, y_B, θ) correspondantes de la plate-forme mobile. Les positions (x_b, y_b) y sont représentées par des croix, et les flèches indiquent la direction de l'avant de la plate-forme mobile. L'essai de validation, effectué sur les robots connus utilisés au chapitre 3, permet d'obtenir le même type de performances sur les tâches pour un suivi en position et sur un suivi en position/orientation. L'extension à la recherche complète des consignes (considérant le cas avec manœuvre) ne pose aucun problème, et les configurations de départ retenues sont telles que la plate-forme mobile soit globalement parallèle à la trajectoire désirée et dans la même direction. L'erreur produite sur les variables opérationnelles de départ (position et orientation de l'organe terminal) sont du même ordre que celles obtenues au chapitre précédent (tableau 4.7).

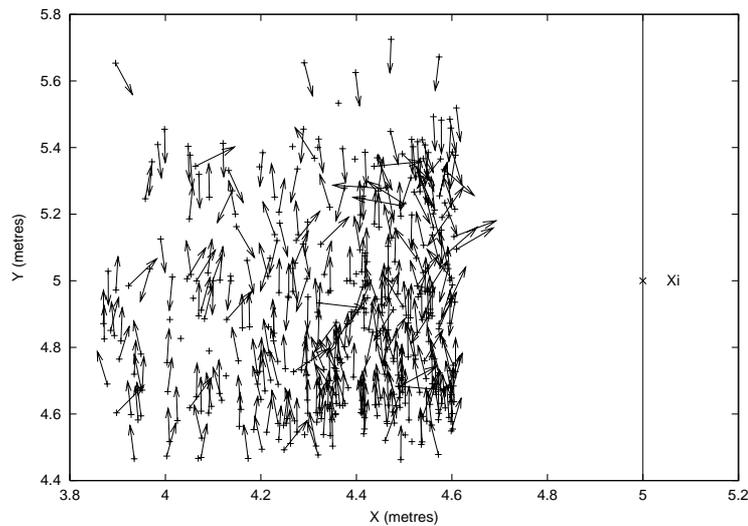


FIG. 5.5: Configurations validées de la plate-forme mobile à la base du manipulateur

La recherche d'une configuration initiale s'effectue indépendamment de celle des vecteurs de consignes. L'optimisation est faite de telle sorte que la valeur d'adaptation du premier vecteur de commande soit maximale, ce qui n'interfère en rien sur la validité de ce vecteur du point de vue de la recherche des consignes. La manipulabilité du robot au cours de cette recherche n'a pas été prise en compte. La configuration initiale retenue en fin de simulation ne peut que favoriser une configuration confortable de manière à minimiser la quantité de vecteurs de consignes différents. Malgré l'indépendance des modules d'optimisation, les deux recherches s'effectuent en parallèle et interagissent et s'influencent en permanence.

L'impact de l'inclusion de ce nouveau module sur la recherche des entrées de commande est essentiellement temporel : les résultats de convergences permettent d'obtenir rapidement le(s) vecteur(s) de consignes qui réalisent le suivi (position ou position/orientation). En revanche, le cas idéal de configuration initiale présenté aux chapitres 2 et 3, est loin d'être le seul testé, puisque nous ne désirons pas introduire de critère subjectif dans l'évaluation (recherche *autonome* des solutions, pas d'influence ni d'intervention de l'utilisateur). Comme nous l'avons dit plus haut, aucune notion de mobilité (ou autre : énergie, ...) n'est introduite. L'aspect temporel de la recherche fait donc l'objet d'une attention particulière dans le paragraphe suivant.

5.3.3 Temps de simulation

La recherche d'un vecteur configurationnel initial s'effectue relativement rapidement (4'10" en moyenne) du fait du faible temps alloué au test des consignes (10 pas de temps, soit 0.01 seconde). Cependant, cette recherche est effectuée pour tous les individus de l'optimisation des entrées de commande et à chaque génération. Le temps additionnel sur une simulation complète, pour un suivi en position/orientation sans manœuvre,

s'élève à 1h40 en moyenne, triplant largement le temps pour la détermination du vecteur de consignes.

De manière à écourter la recherche, les configurations répondant aux contraintes liées à F^{config} (évaluation locale, figure 5.4) sont mémorisées et réintroduites lors de l'initialisation des \tilde{q}_i suivante. Cette manipulation est possible car une même structure est étudiée tout au long de la recherche du vecteur de consignes. L'impact est à double tranchant : en fin de recherche des entrées de commande, d'avantages de vecteurs opérationnels sont à tester, puisque de nouvelles solutions valides apparaissent à chaque génération. Le temps utilisé pour la recherche est directement lié à la quantité de solutions testées, celui-ci sera donc augmenté de par cet aspect. Par contre, une économie est réalisée du côté de la recherche de configurations valables du point de vue des contraintes mécaniques. L'augmentation des possibilités configurationnelles ne présente donc pas directement un intérêt temporel (la perte et le gain s'équilibrent), mais augmente les chances de trouver une bonne configuration dans la base de données ainsi constituée. Les informations qu'elle contient (paramètres de configuration de la plate-forme mobile et du bras manipulateur, valeur d'adaptation pour chaque configuration), ainsi que leur analyse statistique, vont permettre une réelle économie de temps.

L'analyse des solutions testées permet de remarquer que certaines configurations, bien que respectant les contraintes mécaniques, s'avèreront toujours de faible évaluation finale du point de vue du test des consignes. Il est évident, par exemple, qu'un démarrage de la plate-forme perpendiculairement à la paroi offrira des résultats moins bons que si elle démarrerait parallèlement, quel que soit le vecteur d'entrées de commande. Cet aspect est mis en évidence sur l'analyse présentée ci-après, qui résulte de statistiques d'observation du comportement général d'une simulation de recherche des consignes. Les informations fournies par l'observation du comportement des variables intervenant au cours de la simulation ne sont pas toutes directement exploitables : les relations qui peuvent exister entre les variables ne peuvent pas en être extraites. Le cas de θ met très nettement en évidence la remarque effectuée plus haut, c'est-à-dire de meilleurs résultats sur l'évaluation du vecteur des consignes pour $\theta \rightarrow \frac{\pi}{2}$. De ces observations on peut donc prévoir, avant le test des consignes, quel comportement découlera d'une configuration en regardant la valeur de θ . Les informations sur les autres variables, en l'occurrence sur les variables de la chaîne cinématique, ne peuvent être utilisées qu'en connaissant le système étudié (par l'intermédiaire du MGD par exemple). Celui-ci pouvant être de tout ordre par la suite, elles ne seront pas exploitées dans notre cas d'étude.

A partir de ces observations, une contrainte dure supplémentaire est introduite de manière à limiter le test des consignes aux solutions les plus probables. Celle-ci doit

1. définir l'intervalle, lorsqu'il existe, contenant des variables assurant une meilleure évaluation globale,

2. Eliminer du test des consignes les cas proposant des variables configurationnelles hors de cet intervalle.

L'utilisation de cette restriction ne doit cependant limiter en rien l'exploration de l'espace de recherche. Elle ne peut donc pas être mise en œuvre dès l'initialisation (départ de la simulation), et est basée sur la base de données, constituée au fil des générations, précédemment décrite. Son intervention ne doit avoir lieu que dans les cas où cet in-

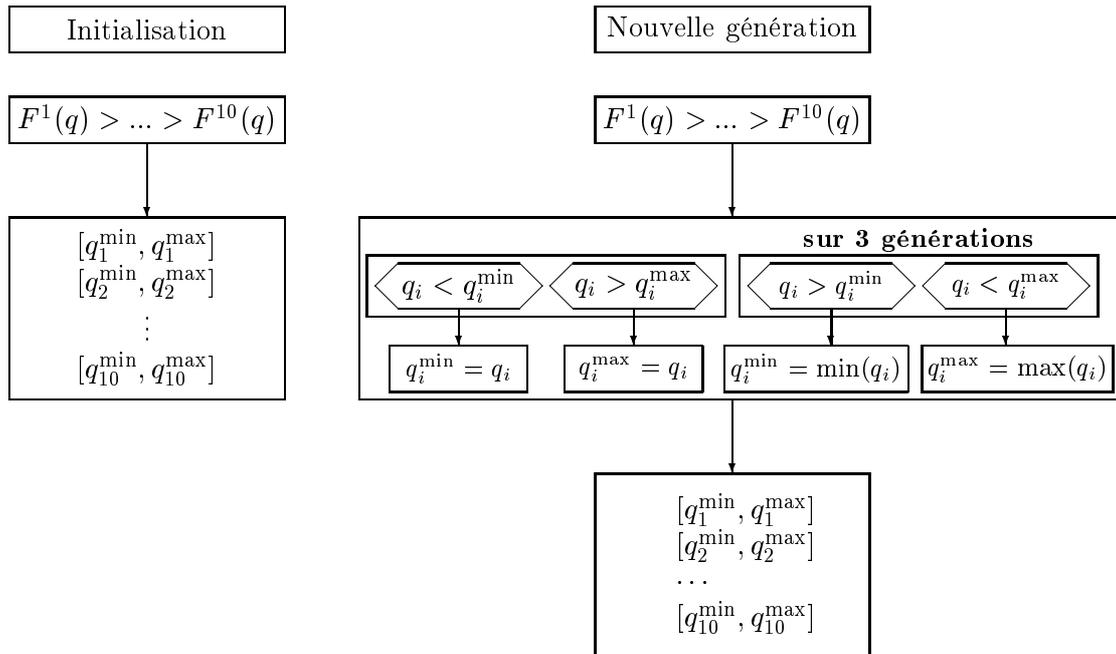


FIG. 5.6: Algorithme pour l'utilisation des performances statistiques de la recherche des configurations initiales.

tervalle peut être détecté sans aucune ambiguïté, de manière à ne pas handicaper la recherche. L'algorithme que nous proposons présente trois étapes principales (figure 5.6) :

1. Classement des individus par ordre de valeurs d'adaptation des configurations croissantes.
2. Etablissement des intervalles, pour chaque variable configurationnelle, basé sur les valeurs minimales et maximales atteintes pour les 10 premiers individus.
3. Si, sur trois générations successives, les bornes trouvées pour un intervalle sont inférieures aux bornes précédentes, remplacement par la valeur maximale obtenue sur ces trois générations. Si une valeur supérieure est trouvée, le remplacement est immédiat et systématique (agrandissement de l'intervalle).

Lorsque ces intervalles sont constitués, seules les solutions configurationnelles les respectant sont testées par les consignes. Ces opérations sont effectuées dès que la base de données contient 20 solutions différentes (i.e., dès la première ou seconde génération). L'inertie qui existe pour la réduction des intervalles permet de ne pas hater une sélection trop stricte, et laisse l'ouverture à de nouvelles possibilités.

L'économie de temps réalisée fait passer de 1h40 à 48 mn en moyenne la recherche de configurations initiales. Les intervalles considérés en fin de simulation sont présentés dans le tableau 5.3.3. Ils confirment les remarques effectuées sur l'analyse des résultats présentés sur la figure 5.7 (très faible intervalle résultant pour θ , intervalles réduits pour x_b et y_b). Les intervalles restant, pour les articulations du manipulateur, mettent

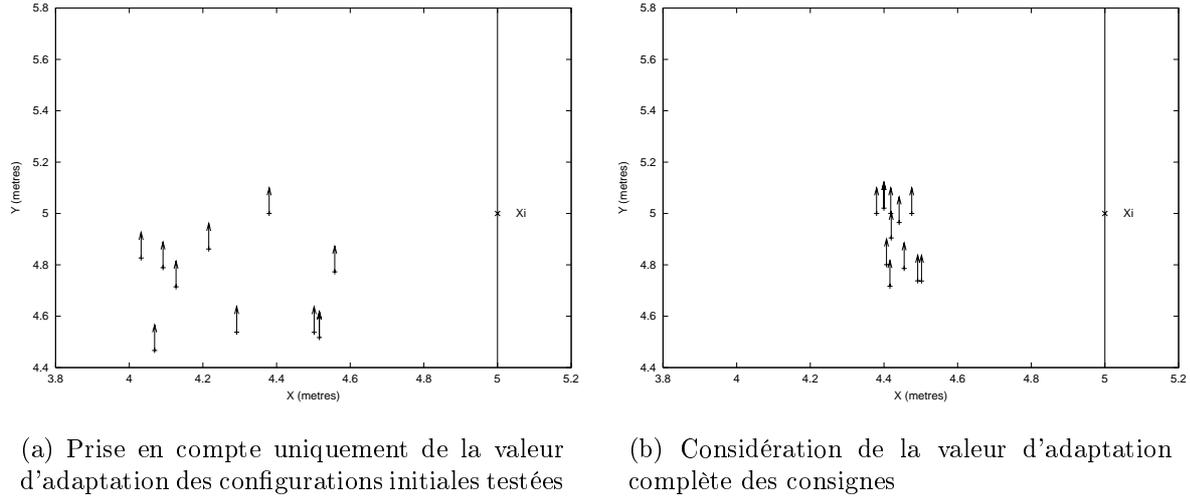


FIG. 5.7: 10 meilleures configurations de la plate-forme mobile, qui définissent les intervalles des configurations testées par U_1 .

en évidence la neutralité de la contrainte puisque les intervalles associés ne suppriment aucune des solutions configurationnelles valables avec des valeurs spécifiques de ces articulations (toutes les valeurs proposées sont comprises dans les intervalles résultants).

q_i	q_{\min}	q_{\max}
x_b	4,379	4,502
y_b	4,717	5,021
θ	$\pi/2$	$\pi/2$
ϕ_1	-0,318	-1,052
ϕ_2	0,318	0,747
ϕ_3	0,256	0,98

TAB. 5.2: Intervalles contenant les configurations initiales testées par les consignes dans le cas du suivi en position

Nous avons donc atteint l'objectif de cette étude, qui était de réduire le temps alloué à la recherche d'une configuration initiale. La contrainte utilisée à cet effet reste objective face à la recherche des consignes, et évolue avec celle-ci, supprimant au fil

des générations les tests des entrées de commande partant de configurations peu probable (statistiquement moins adaptées que d'autres). Cet "apprentissage" de la part de l'algorithme se montre particulièrement efficace : le temps de recherche est divisé par deux. Son utilisation nécessite cependant une connaissance des variables manipulées, de manière à déterminer lesquelles seront étudiées.

5.3.4 Conclusions sur le générateur de consignes

Nous avons mis en place un générateur de consignes fiable, dont les caractéristiques suivantes ont été mises en évidence progressivement :

- Emergence systématique d'une solution
- Convergence rapide
- Stabilité face à une modification de la dimension de la structure
- Réalisation d'un suivi en position et orientation par l'organe terminal du manipulateur mobile
- Détermination d'une configuration initiale permettant de réaliser un suivi optimal du point de vue de la quantité de vecteurs de consignes différents

Les robots utilisés ont servi d'intermédiaires entre des valeurs de paramètres et leur évaluation. Aucun des critères utilisés, dans les évaluations successives, ne sont spécifiques aux modèles utilisés. Cette utilisation des robots comme outil dans l'approche que nous avons adopté permet de conserver un caractère générique au générateur de consignes. De plus, sa fiabilité face à des modifications des exigences de la tâche montrent sa capacité d'adaptation à la résolution d'autres problèmes. Pour l'objectif actuel, ce caractère générique de l'analyse autorise l'utilisation du générateur de consignes sur n'importe quelle structure cinématique. La sélection finale peut avoir lieu, nous en discutons ci-après.

5.4 Sélection finale

La sélection finale utilise tous les algorithmes mis en place précédemment, son cheminement est illustré par la figure 5.8. L'évaluation de la structure se base essentiellement sur la valeur d'adaptation issue de la génération de consignes. C'est en effet celle-ci qui permet de valider une solution structurelle par rapport à la réalisation de la tâche. Il faut toutefois prendre en compte un critère supplémentaire qui permette de distinguer deux solutions qui parviennent à l'objectif. Ce choix est basé sur le caractère minimal de la structure. La fonction d'adaptation utilisée s'écrit alors :

$$F = \frac{1}{n_b} \times P$$

où P est le critère de poids défini au chapitre 4. Le coefficient $1/n_b$ permet de favoriser les structures de ddl inférieur.

Les paramètres évolutionnaires sont fixés en fonction de la dimension de la structure cinématique. Les valeurs sont basées sur la structure PUMA 560 prises en référence

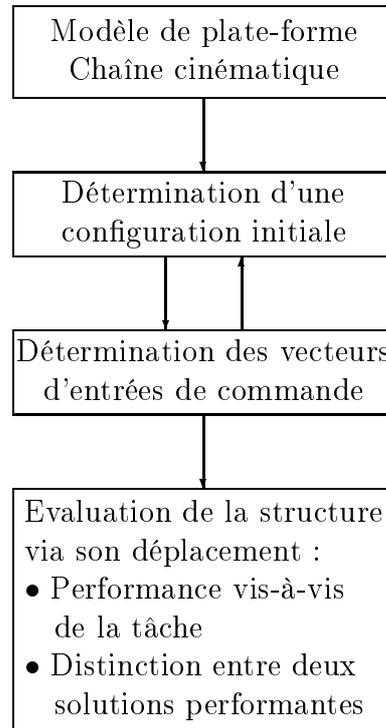


FIG. 5.8: Cheminement jusqu'à la solution finale.

(tableau 5.1). Les caractéristiques structurelles de la solution retenue et les vecteurs d'entrées de commande associés à son déplacement sont présentés dans le tableau 5.3. C'est une structure à 3 degrés de libertés pour la chaîne cinématique, comprenant en plus une liaison fixe. Le robot réalise le suivi dans un mouvement continu et complémentaire des deux composants robotiques (la plate-forme et le manipulateur). Afin de gérer les exigences en orientation, le robot s'arrête quasiment aux angles de la marche pour positionner convenablement son effecteur. La rotation de celui-ci est assurée sur une petite portion de trajectoire, une rotation sur place ne pouvant être assurée avec la définition des paramètres que nous avons adopté. En effet, cela équivaldrait à l'application de consignes sur une distance nulle. Ces résultats mettent en évidence la fiabilité de l'approche et l'abstraction complète du système étudié, et permet une grande adaptabilité sur une recherche sur d'autres type de structures. A titre de comparaison, la simulation complète a été testée sur la tâche partielle (en position et orientation perpendiculaire au mur) décrite au chapitre 2. L'élimination préliminaire se base dans ce cas à une seule sous-tâche en X_0 . La structure sélectionnée, outre le véhicule, est constituée de deux barres soudées en angle droit, ce qui lui permet d'assurer la perpendicularité de l'effecteur à la paroi.

	Véhicule		Chaîne cinématique				$\delta\ell_i$
	R Droite	R Gauche	1	2	3	4	
Orientation (axe)	-	-	\vec{z}	\vec{y}	\vec{x}	\vec{z}	-
Type (liaison)	-	-	R	F	P	R	-
Longueur (barre)	-	-	0,4375	0,1875	0,4375	0,0625	-
U_1	4,87	4,87	0	-	0	0	3,82
U_2	2,12	0,83	0,94	-	-0,24	1,12	0,17
U_3	0,13	0,06	1,83	-	-0,24	1,12	0,01
U_4	1,3	2,56	-1,42	-	0,92	-2,32	0,5
U_5	0,06	0,15	-0,4	-	-0,21	2,2	0,02
U_6	3,21	3,21	0	-	0	0	4

TAB. 5.3: Variables de conception de la solution retenue

5.5 Conclusions sur la méthodologie générale

Nous apportons dans cette partie des éléments de conclusion relatifs à l'ensemble de la méthodologie. Ils effectuent un récapitulatif et des commentaires sur trois aspects du travail effectué :

- les caractéristiques des différents niveaux d'optimisation de la recherche
- les temps de simulation
- l'outil évolutionnaire pour la conception robotique

5.5.1 Optimisations successives

La méthodologie générale est basée sur une succession d'optimisations. Celles-ci permettent l'élimination ou la sélection de solutions sur les quatre niveaux de la problématique :

- l'élimination préliminaire,
- la génération des consignes,
- la recherche d'une configuration initiale,
- la sélection finale.

Le tableau 5.4 effectue un récapitulatif des caractéristiques d'évaluation pour chaque algorithme mis en place. Celui-ci comprend

- l'objectif de la recherche : les sorties de l'optimisation,
- les pré-requis pour effectuer la recherche : les entrées de l'optimisation,
- le détail des boucles d'optimisation, en un ou deux étages, caractérisées par leur tendance éliminatoire, sélective, locale,...
- Les critères utilisés pour l'évaluation, traduisant des contraintes sélectives strictes (généralement liées à la réalisation de la tâche) et des contraintes souples (équilibre et améliorations).

La séparation de la fonction d'adaptation en deux fonctions d'évaluations au sein d'une même recherche s'est avérée nécessaire pour la rigueur de la sélection. En effet, un éva-

Niveau	Évaluation	Critères	Entrées	Sorties
Élimination préliminaire	Configuration (éliminatoire)	<ul style="list-style-type: none"> • Erreur en position • Erreur en orientation • Évitements de collisions 	<ul style="list-style-type: none"> • Modules : représentation, règles d'assemblage • Sous-tâches caractérisant la tâche globale 	<ul style="list-style-type: none"> • Chaînes cinématiques localement validées • Une configuration associée
	Topologie	<ul style="list-style-type: none"> • Mobilité max à l'effecteur • Structure min 		
Génération de consignes	Locale	<ul style="list-style-type: none"> • Erreur en position • Erreur en orientation • Évitements de collisions • Avancement sur la traj. 	<ul style="list-style-type: none"> • Configuration initiale • Tâche complète • Borne sup des consignes • Nombre de ddl • Modèle cinématique pour la simulation 	Vecteurs d'entrées de commande : <ul style="list-style-type: none"> • quantité • Temps d'application • Valeurs des consignes
	Globale	<ul style="list-style-type: none"> • Évaluation locale min • Quantité min de vecteurs de consignes différents 		
Configuration initiale	Partielle (sélective)	<ul style="list-style-type: none"> • Erreur en position • Erreur en orientation • Évitements de collisions 	<ul style="list-style-type: none"> • Modèle complet pour la simulation • Direction désirée • Vecteur de consignes initial 	<ul style="list-style-type: none"> • Vecteur initial des coordonnées configurationnelles
	Complète	<ul style="list-style-type: none"> • Étape partielle validée • Direction • Évitements de collisions • Erreur en situation 		
Sélection finale		<ul style="list-style-type: none"> • Évaluation des consignes max • Structure min 		

TAB. 5.4: Récapitulatif des caractéristiques des différents niveaux d'optimisation constituant la méthodologie générale

luation comprenant un grand nombre de critères s'en trouve dégénérée par l'équilibre qui se crée entre ceux-ci. Si la précision face à la réalisation de la tâche s'en trouve grandement améliorée, cela se produit au détriment du temps et de la simplicité. La difficulté principale pour la mise en place de la méthodologie a donc consisté en l'adaptation de l'interprétation d'un problème de manière à ce que sa résolution reste simple sans perdre de son objectivité. Aussi un des avantages majeurs est la neutralité des évaluations face au système étudié. En effet, celle-ci est concentrée sur la réalisation de la tâche, sans contraintes supplémentaires (souvent subjectives), excepté les critères liés à la structure et à dextérité que nous avons introduit. La conséquence directe est la conservation de la généralité de l'approche face aux solutions recherchées. Elle suppose par contre une connaissance très précise de l'objectif, et un cahier des charges particulièrement soigné.

Les différents cycles d'optimisation sont tous indépendants, à l'exception de la recherche de configuration initiale, intégrée au générateur de consignes (figure 5.8) dans le sens où les interactions entre ces deux modules influencent leurs recherches respectives pendant l'évolution. L'aspect des configurations préférentielles pour la réalisation de la tâche n'a pas été abordé. La configuration initiale influence directement cet aspect et c'est lors de la détermination de celle-ci qu'une contrainte supplémentaire pourrait être introduite. Les consignes déterminées sont donc optimales du point de vue de la configuration initiale sélectionnée.

L'ordre d'utilisation des cycles d'optimisation est essentiel puisque leurs entrées correspondent à, ou comprennent, des éléments liés aux sorties de l'étape précédente (tableau 5.4). La validité du choix des critères a été mise en évidence dans les résultats obtenus, qui correspondent à l'objectif fixé (optimisation essentiellement liée à la tâche).

5.5.2 Performances de l'algorithme

Une analyse partielle des performances de l'algorithme a été effectuée. En effet, l'analyse complète est difficilement réalisable faute de systèmes équivalents de comparaison. Celle-ci est donc partielle dans le sens où les améliorations sont constatées relativement à l'algorithme de base mis en place. Un certain nombre de remarques peuvent être extraites de cette étude : dans un premier temps, l'algorithme se montre stable lors de la modification de la dimension du problème, moyennant l'adaptation des paramètres évolutionnaires (tableau 5.1). Ce cas apparaît lors de l'élimination préliminaire (chromosomes contenant 1 à 6 gènes), la recherche de configurations initiales (3 à 6 gènes) et la détermination des vecteurs d'entrées de commande (5 à 8 gènes). Ainsi, on peut affirmer que les problèmes concernés sont bien posés et les recherches stables en complexité dans les limites de l'étude réalisée. La généralisation n'est cependant pas établie et reste à mettre en place (recherche des chaînes cinématiques à m degrés de liberté, $\forall m$). La recherche des paramètres évolutionnaires optimaux est un travail de longue haleine souvent négligé. Ceux-ci sont effectivement généralement fixés arbitrairement, dans des intervalles assurant une convergence convenable. Les méthodes

évolutionnaires adaptatives sont censées apporter une réponse à ce problème. Elles ont leurs avantages (adaptation des paramètres en fonction de l'évolution en cours) mais n'assurent pas des résultats sur l'ensemble des évolutions traitées par un même algorithme, c'est-à-dire une recherche stable. Elles proposent en fait un échantillonnage de valeurs en fonction de la génération, en espérant trouver l'équilibre entre les paramètres qui fera émerger une solution intéressante. Notre approche, pour la recherche des entrées de commande, a consisté à favoriser un ensemble sélectionné de bonnes solutions au dépend des moins bonnes. L'exploration autour de cet ensemble permet de trouver une solution au problème, quelle que soit l'évolution. Par contre, appliquer cette méthode à la recherche d'une configuration respectant les contraintes géométriques des chaînes cinématiques n'a pas produit de si bons résultats. Si la représentation des solutions respectives est non dégénérée dans les deux cas, les variables liées au chromosome des consignes présentent une redondance que celles du chromosome des configurations n'ont pas. Cette particularité aide la recherche dans le sens où une relation dans la combinaison entre les variables constituant le chromosome permet d'atteindre le même résultat par plusieurs chemins. L'algorithme gère donc le problème de la coopération entre la plate-forme mobile et le bras manipulateur sans que l'on ait fait état d'une quelconque séparation au sein du système.

Les temps de simulations dépendent énormément du problème traité, voire de la qualité des individus (du point de vue de l'évaluation) de la population au sein d'une même évolution. Ceux que nous avons cités au cours de l'étude sont des temps moyens (réalisés au cours des moyennages), mais les variations d'une évolution à l'autre sont très grandes. Des adaptations particulières ont cependant permis de limiter les délais d'attente. Dans le cas de la recherche des consignes par exemple, nous sommes parvenu à les réduire par le regroupement de boucles d'optimisation initialement séparées. La ré-insertion forcée des meilleurs individus, toutes générations confondues, contribue aussi à l'amélioration des performances, du point de vue temporel. L'usage de boucles imbriquées augmente donc considérablement les temps de recherche, mais l'essentiel est utilisé pour la simulation des consignes lors de l'évaluation. L'usage d'évaluations séparées est cependant un choix de qualité vis-à-vis de la précision de la solution (topologique, configurationnelle, commande,...). Le regroupement en une même boucle d'optimisation va donc à l'encontre de nos objectifs. Dans le cas de l'élimination préliminaire, nous désirons obtenir un maximum de topologies différentes avec une précision maximale vis-à-vis des contraintes liées à la tâche. Le regroupement en une seule boucle d'optimisation est réalisable, au détriment de l'un ou l'autre de ces aspects (perte de diversité ou chute de la précision). Les deux niveaux sont donc nécessaires et essentiels pour l'approche. Le cas de la recherche des consignes est différent -bien que menant à la même conclusion- puisqu'on ne peut pas s'acquitter de l'objectif final (atteindre une situation désirée) sans respecter les contraintes de réalisation (en suivant une trajectoire donnée). C'est cet aspect double objectif + contraintes de réalisation qui force la double optimisation.

5.5.3 Commentaires sur l'outils évolutionnaire

Les algorithmes évolutionnaires se sont montrés particulièrement adaptés à l'étude, sur tous les niveaux de la problématique. Les opérateurs utilisés étaient non spécifiques à l'étude, laissant une grande ouverture sur les champs de recherche ultérieurs. Plusieurs remarques peuvent être faites sur les résultats. Les aspects positifs convergent vers l'adaptabilité de la méthode :

- Dans tous les cas d'utilisation, des solutions valables ont été extraites d'un espace de recherche très vaste.
- La non-spécificité des opérateurs au problème traité (on utilise les mêmes pour toute l'étude) vont dans le sens de l'adaptabilité et de la généricité.
- Il a été fait totalement abstraction du système étudié. Cet aspect va aussi dans le sens de la généricité et de l'objectivité de l'évaluation.
- L'établissement de la fonction objective d'évaluation s'est fait sans soucis de contraintes mathématiques à respecter par cette fonction. L'expression des critères la constituant a pu se réduire à sa forme la plus simple.

Les aspects négatifs relèvent essentiellement la mise en place de l'étude, autant pour la pose du problème que le temps nécessaire pour l'effectuer :

- La pose du problème est très délicate à accomplir, dans le sens où il doit être réduit à sa plus simple expression.
- La gestion des paramètres évolutionnaires est très fastidieuse pour un résultat optimal.
- la précision peut être améliorable par le couplage avec une autre méthode d'optimisation. L'équilibre exploration/exploitation est mal géré par l'algorithme, surtout quand des solutions très précises sont attendues.

La généricité se paye par une nécessité de limiter le problème à son expression la plus simple. Ceci explique probablement la stabilité de l'algorithme face à un changement de dimension du problème ("génétiquement bien posé"). Les simplifications sont loin d'être évidentes, ou même réalisables, pour tous types de problèmes. Le bilan général est toutefois très satisfaisant et la technologie évolutionnaire s'est montrée performante et convaincante face aux problèmes de haute complexité.

Conclusion Générale

Méthodologie de la conception orientée tâche

Ce document traite de l'utilisation des déplacements d'un manipulateur mobile dans l'évaluation de sa structure lors de l'étape de conception robotique préliminaire. Dans cet objectif, une méthodologie de la conception orientée tâche utilisant des optimisations successives par les algorithmes génétiques a été mise en place en utilisant deux étapes :

1. une sélection préliminaire évaluant les chaînes cinématiques seules correspondant à une étude restreinte de faisabilité, réalisée en statique et n'utilisant pas les déplacements de la plate-forme mobile.
2. la sélection finale évaluant la capacité des solutions robotiques complètes à effectuer la tâche globale.

La première étape est destinée à limiter l'analyse finale aux seules solutions potentiellement capables d'effectuer la tâche globale dans un objectif d'économie de temps de calcul. L'étape finale a nécessité la mise en place d'un générateur de consignes s'adaptant aux variations topologiques et cinématiques des robots contrôlés. L'approche propose un algorithme qui n'utilise en entrée que le nombre d'actionneurs à contrôler, hormis les informations concernant la tâche à réaliser. Ce module de synthèse d'entrées de commande inclut la détermination de la configuration initiale des robots étudiés.

Contributions

Le travail d'analyse et de synthèse effectué pour la mise en place de cette méthodologie a nécessité des adaptations importantes qui peuvent être considérées comme les contributions apportées. Celles-ci sont :

- La synthèse de consignes en boucle ouverte par les algorithmes génétiques.
- L'adaptation d'un opérateur de croisement autorisant la manipulation de chromosomes de tailles différentes.
- Des simplifications permettant de réduire les informations à leur représentation minimale.

Synthèse d'entrées de commande

Le module de synthèse de consignes permet de déterminer de manière fiable les entrées de commande à appliquer sur chaque actionneur du robot étudié. L'avantage de ce module est son indépendance avec la cinématique du robot en question. L'abstraction faite de la structure permet de baser la gestion de l'évolution des configurations sur la seule évaluation du résultat du déplacement (la réalisation de la tâche). En résulte une abstraction parallèle des problèmes inhérents aux manipulateurs mobiles (gestion de la redondance, coopération entre la plate-forme mobile et le bras manipulateur, équilibre de la structure).

Croisement de chromosomes de longueurs variables

La manipulation par les opérateurs génétiques de chromosomes de longueurs différentes et contenant des informations dont la nature du codage diffère est du plus grand intérêt pour la suppression des boucles d'optimisation jusque là nécessairement imbriquées, ou traitées séparément. Le génotype de l'individu devient une bibliothèque à proprement parler et l'optimisation résultante ne s'en retrouve pas lésée.

Représentation minimale des informations

L'expression des informations sous leur forme minimale est présente tout au long de l'approche. Elle permet la concentration de la recherche essentiellement sur la réalisation de la tâche. Elle apparaît à travers l'évaluation (à tous les niveaux de la problématique), l'abstraction de la structure pour la détermination des consignes (permettant la généralité du module), la non estimation de l'espace de travail (supprimant la recherche de trois paramètres supplémentaires pour le problème plan que nous avons traité).

Perspectives

L'objectif principal de ces adaptations était dirigé vers la détermination d'une réponse adaptée au problème posé, la fiabilité des solutions apportées ainsi que la performance temporelle de résolution. Ses conséquences débouchent sur une généralité de l'approche, qui ouvre de nombreuses perspectives :

- Effectuer une analyse complémentaire des performances de l'algorithme face à la manipulation de chromosomes de contenu divers et de taille supérieure à ceux étudiés dans ce manuscrit, ce permettant de vérifier sa stabilité face aux problèmes complexes abordés et d'en envisager d'autres. La détermination systématique des paramètres évolutionnaires ainsi que la suppression de toutes les boucles d'optimisation imbriquées avec gain de temps est l'objectif de cette étude à court terme.
- Aborder la conception du manipulateur mobile complet, comprenant la plate-forme mobile et le bras manipulateur comme une seule entité. Dans la prolongation

de cette étude figure la synthèse de structures et de consignes en environnement quelconque.

- Adapter la recherche des consignes à un véritable module de commande équivalent à une commande en boucle fermée, c'est-à-dire une recherche à chaque pas de temps des vecteurs d'entrées de commande. Application : commande directe d'un système réel et de modèle inconnu *a priori*, basée sur les retours des capteurs. Adaptation sur n'importe quel robot, seul le nombre d'actionneur -et l'objectif- doit être donné.

Bibliographie

- [Adeli94] H. Adeli, *Advances in Design Optimization*, Chapman and Hall, 1994.
- [Ahmadabadi96] M.N. Ahmadabadi, N. Eiji, *A Cooperative Strategy for a Group of Object Lifting Robots*, Proc. of IEEE Int. Conf. on Intelligent Robots and Systems, pp 125-131, 1996.
- [Antonisse89] J. Antonisse, *A new interpretation of schema notation that overturns the binary encoding constraint*, , 1989.
- [Arai96] T. Arai, J. Ota, *Let us Work Together -Task Planning of Multiple Mobile Robots-*, Proc. of IEEE Int. Conf. on Intelligent Robots and Systems, pp 298-303, 1996.
- [Arora95] J.S. Arora, O.A. Elwakeil, A.I. Chahande, *Global Optimization Methods for Engineering Applications : a review*, Structural Optimization, pp 137-159, 1995.
- [Back96] T. Bäck, *Evolutionary Algorithms in Theory and Practice*, Oxford University Press, 1996.
- [Bonitz97] R.G. Bonitz, T.C. Hsia, *Calibrating a Multi-Manipulator Robotic System*, IEEE Robotics & Automation Magazine, pp 18-22, March 1997.
- [Cellier96] L. Cellier, *Analyse Expérimentale de Schémas de Commande Position/Force d'un Manipulateur Monté sur un Véhicule*, Thèse de doctorat de l'Université de Montpellier, 1996.
- [Cerf94] R. Cerf, *Une théorie asymptotique des algorithmes génétiques*, Thèse de doctorat, Univ. Montpellier II, 1994.
- [Chapelle02] F. Chapelle, , Thèse de doctorat de l'Université Pierre et Marie Curie, 17 septembre 2002.
- [Chedmail96] P. Chedmail, E. Ramstein, *Robot Mechanism Synthesis and Genetic Algorithms*, Proceedings of ICRA'96, IEEE, 1996.
- [Chedmail98] P. Chedmail, E. Dombre, P. Wenger, *La CAO en Robotique*, Hermes, 1998.
- [Chen96] I.M. Chen, J. Burdick, *Determining Task Optimal Modular Robot Assembly Configurations*, Proceedings of ICRA'96, IEEE, 1996.
- [Chen97] M.W. Chen, A.M.S. Zalzal, *Dynamic Modelling and Genetic-based Trajectory Generation for Non-holonomic Mobile Manipulators*, Control Eng. Practice, **5**(1), pp 39-48, 1997.
- [Chirehdast94] M. Chirehdast, H.C. Gea, N. Kikuchi, P.Y. Papalambros, *Structural Configuration Examples of an Integrated Optimal Design Process*, Journal of Mechanical Design, **116**(December), pp 997-1004, 1994.

- [Chocron00] O. Chocron, *Conception Evolutionnaire de Systèmes Robotiques*, Thèse de doctorat de l'Université Pierre et Marie Curie, 27 janvier 2000.
- [Cholvy83] L. Cholvy, J. Foisseau, *Rosalie : A CAD Object-Oriented and Rule-Based System*, Information Processing 83, Mason, REA, pp 501-505, 1983.
- [Craig86] J. Craig, *Introduction to Robotics, Mechanics and Control*, Addison-Wesley, 1986.
- [Darwin59] C. Darwin, *De l'origine des espèces par la sélection naturelle*, Angleterre, 1859.
- [Denavit55] J. Denavit, R.S. Hartenberg, *A kinematic Notation for lower pair Mechanisms based on Matrices*, Journal of Applied Mechanics, vol. 22, 1955.
- [Desai97] J.P. Desai, V. Kumar, *Optimal Motion Plans for Cooperating Nonholonomic Mobile Manipulators in an Environment with Obstacles*, GRASP Laboratory Techn. report 403, 1997.
- [Dickson95] W.C. Dickson, R.H. Cannon Jr., *Experimental Results of Two Free-Flying Robots Capturing and Manipulating a Free-Flying Object*, IEEE, pp 51-58, 1995.
- [Dixon76] L.C.W. Dixon, J. Gomulka, S.E. Hersom, *Optimization in action*, London Academic Press, Chap. Reflections on the global optimization problem, 1976.
- [Dufflo96] M. Dufflo, *Algorithmes stochastiques*, Springer, Paris, 1996.
- [Ferber95] J. Ferber, *Les systèmes multi-agents : vers une intelligence collective*, Inter Edition, 1995.
- [Finger89] S. Finger, J.R. Dixon, *A Review of Research in Mechanical Engineering Design. Part I : Descriptive, Prescriptive and Computer-Based Models of Design Processes*, Research in Engineering Design, 1(1), pp 51-67, 1989.
- [Fisher96] C. Fisher, M. Buss, G. Schmidt, *Soft Control of an Effector Path for a Mobile Manipulator*, Wac'96, 1996.
- [Fogel66] L.J. Fogel, A.J. Owens, M.J. Walsh, *Artificial intelligence through simulated evolution*, New York : John Wiley, 1966.
- [Foulon97] G. Foulon, J.-Y. Fourquet, M. Renaud, *On Coordinated Tasks for Nonholonomic Mobile Manipulators*, 5^{me} IFAC Symposium on Robot Control (SYROCO'97), pp 491-498, 1997.
- [Foulon98] G. Foulon, *Génération de Mouvements Coordonnés pour un Ensemble Constitué d'une Plate-forme Mobile à Roues et d'un Bras Manipulateur*, Thèse de doctorat de l'Institut National des Sciences Appliquées de Toulouse, LAAS, 1998.
- [Fukuda97] T. Fukuda, T. Kaga, *Distributed Decision Making of Dynamically Reconfigurable Robotic System*, Proc. of IEEE Int. Conf. on Intelligent Robots and Systems, pp 1604-1609, 1997.
- [Gascuel96] O. Gascuel & all, *Supervised Classification : a comparison of twelve numerical symbolic and hybrid methods*, Int. Journal of Pattern Recognition and Artificial Intelligence, 1996.
- [Gero85] J.S. Gero, R.D. Coyne, *Knowledge-Based Planning as a Design Paradigm*, In : Design Theory in Computer-Aided Design, IFIP, 1985.
- [Glover77] F. Glover, *Heuristics for Integer Programming Using Surrogate Constraints*, Decision Sciences, 8(1), pp 156-166, 1977.
- [Goldberg89] D.E. Goldberg, *GENETIC ALGORITHMS in search, optimization and machine learning*, Addison Wesley, 1989.

- [Griewank81] O.A. Griewank, *Generalized descent for global optimization*, Optimization Theory Application, 1981.
- [Grefenstette86] J. Grefenstette, *Optimization of Control Parameters for Genetic Algorithms*, IEEE Trans. on Systems, Man and Cybernetics, 1986.
- [Hajek88] B. Hajek, *Cooling schedules for optimal annealing*, Math. of Operations Research, pp 311-329, 1988.
- [Hocaoglu01] C. Hocaoglu, A.C. Sanderson, *Planning Multiple Paths with Evolutionary Speciation*, IEEE Transactions on Evolutionary Computation, **5**(3), pp 169-191, June 2001.
- [Huang97] Q. Huang, S. Sugano, K. Tanie, *Stability Compensation of a Mobile Manipulator by Manipulator Motion : Feasibility and Planning*, Proc. of IEEE Int. Conf. on Intelligent Robots and Systems, pp 1285-1292, 1997.
- [Huang94] Q. Huang, S. Sugano, I. Kato, *Stability Control for a Mobile Manipulator using a Potential Method*, Proc. of IEEE Int. Conf. on Intelligent Robots and Systems, pp 839-846, 1994.
- [Holland75] J.H. Holland, *Adaptation in natural and artificial systems*, Ann Arbor : The University of Michigan Press, 1975.
- [Joskowicz88] L. Joskowicz, S. Addanki, *From Kinematics to Shape : An Approach to Innovative Design*, 7th National Conf. on Artificial Intelligence AAAI, Minneapolis, 1988.
- [Juster85] N.P. Juster, *The Design Process and Design Methodologies*, Tech rept University of Leeds, 1985.
- [Kazuhiro95] S. Kazuhiro, J. Mark, *Automated Optimal Design of Mechanical Conformational Switches*, Artificial Life, **2**(2), pp 1-24, 1995.
- [Khatib96] O. Khatib, K. Yokoi, K. Chang, D. Ruspini, R. Holmberg, A. Casal, *Vehicule/Arm Coordination and Multiple Mobile Manipulator Decentralized Cooperation*, Journal of Robotic Systems, **13**(11), pp 755-764, 1996.
- [Khatib95] O. Khatib, *Inertial Properties in Robotic Manipulation : An Object-Level Framework*, The Int. Journal of Robotics Research, **13**(1), pp 19-36, February 1995.
- [Kim92] J.O. Kim, P. Khosla, *A Multi-Population Genetic Algorithm and its Application to Design of Manipulators*, IROS'92, IEEE, 1992.
- [Kirkpatrick83] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, *Optimization by simulated annealing*, Science *n*°220, pp 671-680, 1983.
- [Kosuge96] K. Kosuge, T. Oosumi, *Decentralized Control of Multiple Robots Handling an Object*, Proc. of IEEE Int. Conf. on Intelligent Robots and Systems, pp 318-323, 1996.
- [Koza92] J.R. Koza, *Genetic Programming : on the Programming of Computers by Means of Natural Selection*, The MIT Press, 1992.
- [Lee96] J. Lee, F. Lein, T. Wei, *Task-Based Specifications Through Conceptual Graphs*, AI for Applications, **August**, pp 60-70, 1996.
- [Leger99] C. Leger, *Automated Synthesis and Optimization of Robot Configurations : an Evolutionary Approach*, Phd thesis, Carnegie Mellon University, December 1999.
- [Libardi88] E.C. Libardi, J.R. Dixon, M.K. Simmons, *Computers Environments for the Design of Mechanical Assemblies : A Research Review*, Engineering with Computers, pp 121-136, 1988.

- [Macready95] W.G. Macready, D.H. Wolpert, *What Makes an Optimization Problem Hard?*, Tech. rept., Santa Fe Institute, 1995.
- [Maher87] M.L. Maher, *A Knowledge-Based Approach to Preliminary Design Synthesis*, Tech. rept. Engineering Design Research Center, Carnegie Mellon University, 1987.
- [Martin95] E. Martin, E.G. Papadopoulos, J. Angeles, *On the Interaction of Flexible Modes and On-off Thrusters in Space Robotic Systems*, IEEE, 1995.
- [Matsumaru95] T. Matsumaru, *Design and Control of the Modular Robot Systems : TOMMS*, Proceedings of ICRA '95, IEEE, 1995.
- [Mendel65] G. Mendel, *Experiments in plant Hybridization*, 1865.
- [Michalevicz92] Z. Michalevicz, *Genetic Algorithms + Data Structures = Evolution Programs*, Springer-Verlag, 1992.
- [Minoux83] M. Minoux, *Programmation mathématique, théorie et algorithmes*, Dunod et Bordas, 1983.
- [Miyata96] N. Miyata, J. Ota, T. Arai, E. Yoshida, D. Kurabayashi, J. Sasaki, Y. Aiyama, *Cooperative Transport with Regrasping of Torque-Limited Mobile Robots*, Proc. of IEEE Int. Conf. on Intelligent Robots and Systems, pp 304-309, 1996.
- [Nassal96(2)] U.M. Nassal, *Motion coordination and Reactive Control of Autonomous Multi-Manipulator Systems*, Journal of Robotic Systems, **13**(11), pp 737-754, 1996.
- [Nassal96] U.M. Nassal, R. Junge, *Fuzzy Control for Mobile Manipulation*, Proc. of IEEE Int. Conf. on Robotics and Automation, pp 2264-2269, 1996.
- [Obayashi97] S. Obayashi, *Pareto Genetic Algorithm for Aerodynamic Design Using the Navier-Stokes Equations*, Genetic Algorithms and Evolution Strategies in Engineering and Computer Sciences, pp 245-266, 1997.
- [Ohkawa98] K. Ohkawa, T. Shibata, K. Tanie, *Method for Generating of Global Cooperation Based on Local Communication*, Proc. of IEEE Int. Conf. on Intelligent Robots and Systems, 1998.
- [Osumi96] H. Osumi, *Cooperative Strategy for Multiple Mobile Manipulators*, Proc. of IEEE Int. Conf. on Intelligent Robots and Systems, pp 554-559, 1996.
- [Ouedzou99] F. Ben Ouedzou, *Méthodes de Conception et Outils de Simulation de Systèmes Robotiques*, Habilitation à Diriger les Recherches, Janvier 1999.
- [Paljug95] E. Paljug, X. Yun, *Experimental Study of Two Robot Arms Manipulating Large Objects*, IEEE Transactions on Control Systems Technology, **3**(2), pp 177-188, June 1995.
- [Papadopoulos96] E.G. Papadopoulos, D.A. Rey, *A New Measure of Tipover Stability Margin for Mobile Manipulators*, Proc. of IEEE Int. Conf. on Robotics and Automation, pp 3111-3116, 1996.
- [Paredis96] C. Paredis, P. Khosla, *A Rapidly Deployable Manipulator System*, ICRA '96, IEEE, 1996.
- [Perrier97] C. Perrier, P. Dauchez, F. Pierrot, *Towards the Use of Dual Quaternions for Motion Generation of Non-Holonomic Mobile Manipulators*, Proc. of IEEE Int. Conf. on Intelligent Robots and Systems, pp 1293-1298, 1997.
- [Pin96] F.G. Pin, K.A. Morgansen, F.A. Tulloch, C.J. Hacker, K.B. Gower, *Motion Planning for Mobile Manipulators with a Non-Holonomic Constraint Using the FSP Method*, Journal of Robotic Systems **13**(11), pp 723-736, 1996.

- [Pires01] E.J. Solteiro Pires, J.A. Tenreiro Machado, P.B. de Moura Oliveira, *An Evolutionary Approach to Robot Structure and Trajectory Optimization*, ICAR'01, 10th Int. Conf. on Advanced Robotics, pp 333-338, August 2001.
- [Pires00] E.J. Solteiro Pires, J.A. Tenreiro Machado, *Trajectory Optimization for Redundant Robots Using Genetic Algorithm*, GECCO 2000, Proceedings of the Genetic and Evolutionary Computation Conference, p 967, July 2000.
- [Price87] W.L. Price, *Global optimization algorithms for CAD workstation*, Journal of Optimization and Theory Applications, 1987.
- [PUMA560] <ftp://ftp.mathworks.com/pub/contrib/v4/misc/robot/puma560.m>
- [Radcliffe95] N.J. Radcliffe, P.D. Surry *Fitness variance of formæand performance prediction*, Morgan Kaufmann, 1995.
- [Rechenberg65] I. Rechenberg, *Cybernetic solution path of an experimental problem*, Farnborough Hants : Ministry of Aviation, Royal Aircraft Establishment, 1965.
- [Regnier96] S. Regnier, *Une méthodologie Distribuée et Interactive pour l'Aide à la Conception de Systèmes Robotiques*, Thèse de doctorat de l'Université Pierre et Marie Curie, 10 juillet 1996.
- [Rey97] D.A. Rey, E.G. Papadopoulos, *On-line Automatic Prevention for Mobile Manipulators*, Proc. of IEEE Int. Conf. on Intelligent Robots and Systems, pp 1273-1278, 1997.
- [Rinnooy87] A.H.G. Rinnooy, G.T. Timmer, *Stochastic global optimization methods Part I and Part II*, Math. Prog., 1987.
- [Rinnooy85] A.H.G. Rinnooy, C.G.E. Boender, G.T. Timmer, *Computational mathematical programming*, K. Schnittkowski, Chap. A stochastic approach for global optimization, 1985.
- [Ruaux98] P. Ruaux, *Conception, Modélisation et Commandes Dynamiques d'un Robot Mobile Autonome Modulaire. Application à un Environnement non Contraint*, Thèse de doctorat, Univ. Pierre et Marie Curie, 1998.
- [Rudolf97] G. Rudolf, *Convergence Properties of Evolutionary Algorithms*, Hamburg : Kovac, 1997.
- [Saadia00] N. Saadia, Y. Amirat, J. Pontnau, N.K. M'Sirdi, *Neural Hybrid Control of Manipulators, Stability Analysis*, Robotica vol.19, pp 41-51, 2001.
- [Saadia99] N. Saadia, Y. Amirat, K. Djouani, J. Pontnau, *Commande Neuronale Adaptive pour le Contrôle en Effort des Robots Compliantes*, Journal Européen des Systèmes Automatisés, **33**(1-2), 1999.
- [Sakka01] S. Sakka, O. Chocron, *Optimal Design, Configurations and Positions For a Mobile Manipulation Task Using Genetic Algorithm*, ROMAN'01 , 10th IEEE Int. Conf. on Robot and Human Communication, pp 268-273, Paris/Bordeaux, France, Sept. 2001.
- [Sakka02] S. Sakka, N.K. M'Sirdi, *Reliability of Optimal Evolutionary-Based Control for a Mobile Manipulator Trajectory Following Task*, ISMCR'02, Int. Symp. on Measurement and Control in Robotics, Bourges, France, June 2002.
- [Sakka02(b)] S. Sakka, N.K. M'Sirdi, *Reliable Evolutionary-Based Command Input Generator for a Simple Mobile Manipulation Task*, ROMAN'02 , 11th IEEE Int. Conf. on Robot and Human Communication, Berlin, Germany, Sept. 2002.

- [Sasaki96] J. Sasaki, J. Ota, T. Arai, Y. Aiyama, *Optimal Arrangement for Handling Unknown Objects by Cooperative Mobile Robots*, Proc. of IEEE Int. Conf. on Intelligent Robots and Systems, pp 112-117, 1996.
- [Schmidt95] L.C. Schmidt, J. Cagan, *Recursive Annealing : A Computational Model for Machine Design*, Research in Engineering Design, **1**(7), pp 102-125, 1995.
- [Schoenauer98] M. Schoenauer, Z. Michalewicz, *Sphere Operators and their Applicability for Constrained Parameter Optimization Problems*, LNCS, 1998.
- [Schwefel81] H. Schwefel, *Numerical optimization of computer models*, Chichester : John Wiley, 1981.
- [Sebag97] M. Sebag, M. Schoenauer, C. Ravisé, *Inductive Learning of Mutation Step-size in Evolutionary Parameter Optimization*, LNCS, vol. 1213, 1997.
- [Seraji95] H. Seraji, *Configuration Control of Rover-Mounted Manipulators*, IEEE Int. Conf. on Robotics and Automation, Nagoya, Japan, pp 2261-2266, 1995.
- [Shin95] J.H. Shin, I.K. Jeong, J.J. Lee, W. Ham, *Adaptative Robust Control for Free-Flying Space Robots Using Norm-Bounded Property of Uncertainty*, IEEE, 1995.
- [Soper00] A.J. Soper, C. Walshaw, M. Cross, *A Combined Evolutionary Search and Multilevel Optimisation Approach to Graph Partitioning*, Mathematics Research Report 00/IM/58, University of Greenwich, London, UK, 2000.
- [Sugano93] S. Sugano, Q. Huang, I. Kato, *Stability Criteria in Controlling Mobile Robotic Systems*, Proc. of IEEE Int. Conf. on Intelligent Robots and Systems, pp 832-838, 1993.
- [Tahboub96] K.A. Tahboub, *Robust Control of Mobile Manipulators*, Journal of Robotic Systems, **13**, pp 699-708, 1996.
- [Tahboub97] K.A. Tahboub, *Observer-Based Control for Manipulators with Moving Bases*, Proc. of IEEE Int. Conf. on Intelligent Robots and Systems, pp 1279-1284, 1997.
- [Tomiyama89] T. Tomiyama, T. Kiriya, T. Takeda, D. Xue, H. Yoshikawa, *Metamodel : A Key to Intelligent CAD System*, Research in Engineering Design, **1**(1), 1989.
- [Wang96] Z. Wang, E. Nakano, T. Matsukawa, *Realizing Cooperative Object Manipulation using Multiple Behaviour-based Robots*, Proc. of IEEE Int. Conf. on Intelligent Robots and Systems, pp 310-317, 1996.
- [Wolpert95] D.H. Wolpert, W.G. Macready, *No Free Lunch Theorems for Search*, Tech. rept. Santa Fe Institute, 1995.
- [Yamamoto98] Y. Yamamoto, X. Yun, *Effect of the Dynamic Interaction on Coordinated Control of Mobile Manipulators*, IEEE Transactions on Robotics and Automation, 1998.
- [Yamamoto97] Y. Yamamoto, X. Yun, *A Modular Approach to Dynamic Modelling of a Class of Mobile Manipulators*, Int. Journal of Robotics and Automation, **12**(2), 1997.
- [Yamamoto95] Y. Yamamoto, X. Yun, *Coordinated Obstacle Avoidance of a Mobile Manipulator*, Proc. of IEEE Int. Conf. on Robotics and Automation, pp 2255-2260, 1995.
- [Yamamoto94(a)] Y. Yamamoto, X. Yun, *Coordinating Locomotion and Manipulation of a Mobile Manipulator*, Transactions on Automation Control, **39**(6), 1994.
- [Yamamoto94(b)] Y. Yamamoto, *Control and Coordination of Locomotion and Manipulation of a Wheeled Mobile Manipulator*, Thèse de doctorat de l'Université de Pennsylvanie, Grasp Laboratory, Philadelphie, 1994.

-
- [Yoshikawa80] H. Yoshikawa, *General Design Theory and a CAD System*, Proc. of IFIP pn Man/Machine Communication in CAD/CAM, Tokyo, 1980.
- [Yoshikawa96] T. Yoshikawa, M. Kurisu, T. Kozuka, *Deciding Grasping Positions and Re-grasping Action by Cooperating Multiple Mobile Robots*, Proc. of IEEE Int. Conf. on Intelligent Robots and Systems, pp 118-124, 1996.
- [Yun93] X. Yun, *Object Handling Using Two Arms Without Grasping*, The Int. Journal of Robotics Research, **12**(1), February 1993.

ANNEXES

Annexe A

Etat de l'art en Manipulation Mobile

A.1 Robots manipulateurs mobiles

Un Manipulateur Mobile résulte de l'assemblage d'un bras manipulateur et d'une plate-forme mobile. Une plate-forme mobile seule ne peut intervenir sur l'environnement et un manipulateur se trouve rapidement à la limite de son espace de travail. Cette association robotique présente donc de nombreuses possibilités puisque la complémentarité de deux robots différents à tout point de vue permet l'élargissement du champ d'action de chacun, tant au niveau de la définition ou du degré de complexité de la tâche que de sa réalisation. Les avantages de ce système touchent par exemple à l'évitement des singularités, le contournement d'obstacles, l'augmentation de l'espace de travail, la coopération...

Si ces améliorations sont particulièrement attrayantes, elles sont accompagnées d'une grande complexité de réalisation. Ce chapitre est destiné à montrer les nouvelles possibilités qu'offre ce robot et les difficultés qui leur sont associées. De nombreuses études ont déjà été effectuées sur les bras manipulateurs et sur les engins locomoteurs robotisés et des solutions aux problèmes respectifs ont été apportées. Cependant l'association met en œuvre un très grand nombre de paramètres dont l'étude de la gestion est toujours d'actualité. Quoi qu'il en soit, leur détermination fait systématiquement appel à l'optimisation de critères qui varie avec les impératifs liés à la tâche.

En revanche, le problème de conception de tels robots a encore été très peu étudié. En effet les deux tâches -de manipulation et de déplacement, du point de vue de la conception, sont considérées comme totalement découplées. Il suffit donc, pour obtenir un manipulateur mobile efficace du point de vue de sa structure, d'étudier une base mobile adaptée au déplacement dans l'environnement désiré et un bras manipulateur adapté à la tâche de manipulation.

La commande de ces robots a été beaucoup plus abordée ([Cellier96], [Foulon97], [Khatib96], ...). Sur des manipulateurs et des plate-formes existants -donc "connus"- et réunis, la détermination de chaque paramètre de liaison devient un problème à une infinité de solutions. En effet, placer l'effecteur à un point précis (et avec une orientation précise) peut s'obtenir par une infinité de positions du "corps" du robot. Un tel problème de redondance est bien sûr abordé en fonction de la finalité désirée. Un découplage total des tâches de manipulation et de locomotion permet par exemple une intervention ponctuelle du bras manipulateur (ramassage d'échantillons...).

A.1.1 Bref recadrage historique

Un robot est défini comme une machine capable d'interagir avec son environnement dans le but d'atteindre un objectif qui lui a été signifié. Les robots en tant que tels ont très tôt alimenté la créativité des écrivains de science fiction -lire par exemple les descriptions de l'automate évolué de Villier de L'Isle-Adam dans *l'Eve Future* (1870) ou la destruction du parasite humain par Copek, premier utilisateur du mot "robot", dans *R.U.R.* (1921). Il n'en va pas de même pour l'évolution technologique réelle de nos machines autonomes. En effet leur évolution s'est faite avec et par nos exigences et connaissances, partant du simple outil améliorant nos performances manuelles, passant par des automates capables d'effectuer des gestes simples et répétitifs. C'est l'utilité industrielle de ces machines qui a réellement permis leur développement. Les manipulateurs ont les premiers été envisagés dans le cadre de la manipulation d'objets irradiés dès 1940 avec l'apparition du nucléaire.

Depuis trente ans seulement, l'intérêt s'est réellement focalisé sur les robots et les possibilités qu'ils offrent, tant du point de vue de l'automatisation industrielle que de l'exploration, l'intervention et le service en milieux hostile. L'adaptation à un environnement parfois très fortement contraint a d'abord été exigée, permettant de suivre des consignes strictes malgré des perturbations et incertitudes. L'écriture formalisée des modèles utilisés est alors devenue nécessaire et systématique, amenant naturellement des commandes plus évoluées encouragées par les progrès spectaculaires récents de la technologie informatique. De nos jours des missions complexes d'exploration imposant des prises de décisions importantes et une hausse de leur autonomie sont demandées. Lorsque l'intervention humaine est possible, la téléopération lui autorise une partie ou la totalité de la prise de décision. Elle est cependant souvent rendue difficile du fait des temps de transmissions (exploration spatiale) ou des conditions extrêmes d'utilisation du robot (intervention en milieux irradié, volcanique, sous-marin). L'assistance permanente par les robots dans notre quotidien (transfert de produits, aide en milieux hospitalier, usinage...) impose aussi leur autonomie de décision même si l'entretien technologique de telle machines nécessite une main d'œuvre nouvellement spécialisée.

La famille des manipulateurs mobiles découle de cette évolution. Le premier brevet pour un manipulateur programmable (réalisé par Georges C. Devol) *destiné à l'industrie* à été déposé et exploité au début des années 60. Il faudra attendre les années

70 pour que l'étude des robots mobiles, encouragée par les exploits du spatial, soit d'actualité *dans les laboratoires de recherche*. Développés et améliorés sans cesse et séparément, ces deux robots font leurs preuves dans leurs milieux d'action respectifs. C'est dans l'esprit de complexification des exigences des missions robotiques que l'idée de leur association est naturellement arrivée. En effet, l'union de la manipulation et du déplacement permet de rapprocher les compétences de ce nouveau robot de celles de l'homme et donc d'intervenir à sa manière.

A.1.2 Actualités et perspectives pour les manipulateurs mobiles

Le champs d'utilisation des manipulateurs mobiles est très large, puisqu'il agit dans tout domaine où manipulation et déplacement sont nécessaires. Autant dire partout où l'homme peut ou pourrait intervenir. L'utilisation actuelle des manipulateurs mobiles est très variée et touche de nombreux domaines A.1 :

- Exploration ou intervention en milieux hostiles : spatial, nucléaire, sous-marin, déminage...
- Industrie : transport et manipulation de produits nocifs, prise en charge des travaux physiquement fastidieux voire dangereux, peinture, soudage, nettoyage de vitres...
- Médical : aide aux handicapés
- Agriculture : cueillette et traitement des fruits ou fleurs



FIG. A.1: Manipulateurs mobiles en fonction

La figure A.1 illustre quelques exemples de robots existants. Ces exemples illustrent les cas de l'exploration Marsienne (Carnegie Mellon Robotics Institute) ou sous-marine

jusqu'à 6000m, du ramassage de tomates (LASE, Faculty of Agriculture, Okayama University, Japan) ou de la coopération entre manipulateurs mobiles, utile entre autres pour le transport d'objets volumineux (Robotics Laboratory, Stanford University, EU). Les manipulateurs mobiles existants utilisés dans l'industrie font appel à des structures simplifiées de robots mobiles ou de manipulateurs du fait de la complexité du problème à résoudre. En effet la fiabilité de ces robots reste à valider, ainsi que leur autonomie. La problématique soulevée par la manipulation mobile est abordée ci-après.

A.1.3 Problématique de la manipulation mobile

Les possibilités offertes par cette famille robotique ne font aucun doute. Ses avantages sont essentiellement liés à l'élargissement du champ d'action de chaque robot. La mise en pratique présente quelques difficultés encore d'actualité. Certaines sont évidentes puisque déjà rencontrées lors de l'étude des manipulateurs ou des plate-formes mobiles indépendamment, par exemple, le respect des contraintes de non-holonomie pour la base mobile ou d'évitement des singularités cinématiques pour le bras manipulateur. A ces contraintes "classiques" s'ajoutent de nouveaux problèmes du fait de l'association des deux sous-robots :

- L'augmentation conséquente du degré de redondance du système
- Les interactions dynamiques
- La complémentarité des deux robots
- La stabilité du système

La redondance traduit le fait qu'à une position de l'organe terminal du manipulateur (l'effecteur) correspond une infinité de positions possibles pour le corps du robot. Mathématiquement le modèle du robot n'est pas inversible directement. Ce problème apparaissait déjà dans le cas de certains bras manipulateurs, il est à présent augmenté des degrés de liberté de la base mobile (trois de plus pour un déplacement plan par exemple).

Le problème lié à l'utilisation complémentaire des deux robots rejoint celui de la redondance, à savoir comment utiliser la base mobile pour que l'aide à la manipulation soit effective. Les interactions dynamiques n'interviennent bien sûr que dans le cas d'une étude en dynamique : les accélérations de chaque robot vont créer des forces agissant sur le robot partenaire, ces forces n'étant plus négligeables à partir d'un certain niveau d'accélération. La stabilité et la précision de déplacement de l'effecteur s'en ressentent.

Enfin, la stabilité du système est à considérer comme un problème à part entière lors des déplacements ou de la manipulation d'un objet lourd, afin d'éviter le renversement du manipulateur mobile. Les différentes approches proposées dans la littérature pour l'étude et la résolution de ces problèmes sont décrites ci-après.

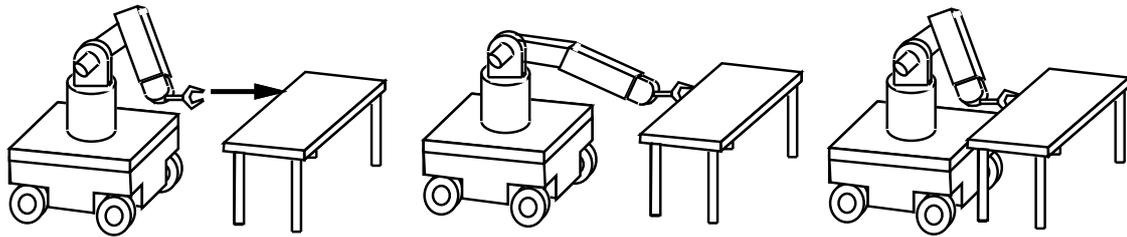


FIG. A.2: Utilisation de la redondance pour déterminer une configuration confortable.

Simplification des modèles

Afin d'aborder le problème complexe de la manipulation mobile, la plupart des auteurs séparent le bras de la base mobile de manière à ne pas augmenter le degré de redondance du système, qui est alors égal à celui du bras manipulateur. Nassal, par exemple, introduit le "concept de la coordination transparente" : il considère une tâche cinématique de manipulation mobile comme l'association de deux sous-tâches assurées par deux contrôleurs individuels (fig. A.2), une pour l'effecteur (sous-tâche destinée au bras), qui assure son positionnement en fonction de la position de la base, et la seconde pour la configuration du bras (sous-tâche destinée à la base), qui assure une configuration optimale dans la réalisation de la tâche de manipulation, et qui n'influence pas les mouvements de l'effecteur. Khatib utilise la même approche en dynamique avec deux structures, "macro" (la base mobile) et "mini" (le bras manipulateur), puis oublie la base en utilisant une plate-forme holonome. Il étudie alors le problème d'un bras manipulateur seul dont la base serait non fixe.

Complémentarité

Une base mobile et un bras manipulateur sont deux robots de comportements très différents, leur association peut donc apporter un progrès considérable dans la gestion de tâches complexes. On distingue trois degrés d'intervention de la plate-forme mobile dans le cadre de l'aide à la manipulation :

- déplacements (intervention) très occasionnels,
- intervention non systématique,
- assistance permanente.

Dans le cas de déplacements très occasionnels de la base, celle-ci n'est utilisée que pour amener le bras à un emplacement optimal par rapport aux différents points à atteindre, et elle est immobile pendant la réalisation de la tâche de manipulation. Le problème revient donc à déterminer la position optimale de la base afin de limiter ses déplacements, le système bras et base mobile est totalement découplé et les études réalisées sur chacun de ces robots peuvent être exploitées. Les interactions dynamiques n'interviennent plus puisque les deux robots ne sont jamais en mouvement en même temps. Par contre, les risques de renversement de la plate-forme doivent être pris en compte pour son placement.

L'intervention non systématique de la plate-forme considère l'aide à la manipulation, mais la base mobile n'intervient que lorsque cela est nécessaire. Donc, si le bras peut manipuler seul, le système se limite à un bras manipulateur fixe. Dans le cas contraire (dépassement de l'espace atteignable), la tâche de la base est à calculer. Les deux robots travaillent alors en parallèle avec une minimisation du travail de la base mobile.

Dans la troisième approche, la base est en perpétuel déplacement de manière à ce que le manipulateur soit constamment en configuration optimale dans la réalisation d'une tâche de manipulation. Le manipulateur est alors considéré comme un système complet et complémentaire, les déplacements des deux robots dépendent l'un de l'autre.

Interactions dynamiques

Les interactions dynamiques résultent des déplacements de masses liés aux mouvements du robot partenaire. Elles peuvent être considérées comme négligeables lorsque ces déplacements se font avec des petites vitesses. Toute étude quasi-statique ou cinématique revient donc à réunir deux robots dans un système complètement découplable et sans modification supplémentaire. L'étude en dynamique, en revanche, présente des différences de modélisation selon la considération d'un manipulateur mobile comme un robot unique avec un modèle global ou comme l'addition des deux modèles, celui du bras et celui de la base.

Une étude approfondie de ces interactions a été menée par Yoshio Yamamoto, dans laquelle il compare les différences au cours d'un même déplacement selon la prise en compte ou non des effets du bras manipulateur sur la base et de la base sur le bras, chargé ou à vide [Yamamoto94(b)]. Ces interactions sont modélisées par un terme additionné au modèle dynamique de chaque robot, le système étant découplé.

Une autre approche est celle de Tahboub ([Tahboub96], [Tahboub97]), qui considère les mouvements de la base comme des perturbations extérieures inconnues sur le bras, perturbations cinématiques (déplacements) et dynamiques (forces appliquées à la base du bras). Il considère en fait que la cible est en mouvements et les mouvements du bras compensent cette erreur pour que l'effecteur reste sur une trajectoire imposée. Son étude se limite au bras, la base étant quelconque et commandée par ailleurs. La trajectoire de l'effecteur dans le repère du monde est ramenée au repère du bras manipulateur. Cette méthode est particulièrement efficace pour prévenir les interactions "modélisables" dans les équations de la dynamique, c'est-à-dire les forces dynamiques résultant des accélérations de la base. Elle l'est en revanche beaucoup moins dans le cas des perturbations non modélisables, au niveau de la cinématique.

Martin et al. [Martin95] se sont particulièrement penchés sur le problème des interactions dans l'objectif d'un robot spatial (fonctionnant hors gravité), donc pour un système où les perturbations liées aux mouvements du bras ou de la base ont un impact particulièrement important sur le robot partenaire. Pour modéliser ces interactions, le

manipulateur mobile est associé à deux masses, la base mobile et la charge utile, reliées par un ressort-amortisseur.

Stabilité

Le problème du renversement a été beaucoup moins exploré du fait de la différence usuelle de poids entre les bases mobiles utilisées et les bras manipulateurs : le renversement est peu probable. Certains auteurs (Huang, Rey) se sont penchés sur le problème particulier de l'équilibre, partant du principe de réduction du besoin de l'espace de manœuvre de la base, ou d'un bras devant porter des objets très volumineux ou lourds (troncs d'arbres, voitures...) tout en restant dans des dimensions raisonnables. Dans le cas où l'équilibre du manipulateur mobile doit être pris en compte, la redondance est cette fois nécessaire pour déterminer l'ensemble des positions acceptables du point de vue de l'équilibre du manipulateur mobile.

Conclusion

Quelle que soit la manière d'aborder les problèmes liés à la manipulation mobile, le nombre de paramètres mis en jeu reste très grand. L'aspect réellement étudié jusqu'à ce jour en manipulation mobile a trait à la commande de ces robots, et toutes les approches font appel à l'optimisation de critères réunis dans une fonction objective (ou fonction coût). Le problème de la manipulation mobile peut s'énoncer comme un problème d'optimisation globale sous contraintes. Nous allons donc aborder le problème de l'optimisation avant de discuter des différentes approches de détermination des configurations optimales faites dans la littérature pour les manipulateurs mobiles.

A.2 Optimisation pour les manipulateurs mobiles

L'utilisation d'un ensemble de critères réunis dans une fonction objective à optimiser est systématique pour résoudre le problème complexe de la manipulation mobile. Après l'introduction de quelques définitions et des paramètres généralement associés aux manipulateurs mobiles, ce chapitre présente les difficultés rencontrées pour les réaliser et les solutions proposées par les auteurs.

A.2.1 Optimisation de la structure

Le problème de la définition d'une structure adaptée à une tâche de manipulation mobile a été très peu abordé. En fait, il est souvent considéré que la mise en place de la structure d'un manipulateur mobile est rigoureusement la même que la construction d'un bras manipulateur et d'une base mobile séparément, et les travaux abondent pour ces deux structures. On considère ici que chacun de ces robots ont des tâches parfaitement découplées, la structure locomotrice étant adaptée à l'environnement de déplacement et le bras à la manipulation. La recherche dans le domaine de la manipulation mobile à proprement parler s'est donc concentrée sur la gestion des problèmes

de la réalisation d'une tâche quels que soient les robots utilisés, dans la plupart des cas des bras manipulateurs classiques existants, pareil pour les bases mobiles. La suite de ce chapitre traite donc des différentes approches utilisées pour déterminer une configuration optimale d'un manipulateur mobile en fonction de la tâche donnée.

A.2.2 Notations et définitions générales

Les définitions et notations suivantes seront utilisées par la suite, et les citations d'autres auteurs seront traduites de manière à leur rester fidèles. D'une manière générale, l'indice b est lié à la base mobile, m au bras manipulateur et mm au manipulateur mobile.

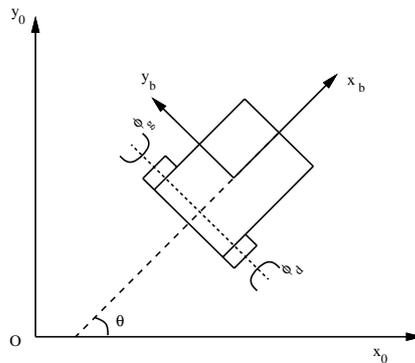


FIG. A.3: Paramètres de la base dans le référentiel fixe.

Les **coordonnées opérationnelles** expriment la **situation** du robot dans le repère du monde, c'est-à-dire la position d'un point fixe du robot et l'orientation du repère attaché à ce point par rapport au repère du monde. Elles s'expriment par le vecteur coordonnées opérationnelles $X = (x_1, \dots, x_{n_o})$ où n_o est la dimension de l'espace opérationnel (3 en bidimensionnel ou 6 en tridimensionnel). Si on considère le bras manipulateur, le repère de référence est celui qui lui est attaché. Ses coordonnées opérationnelles correspondent à la position et l'orientation de l'effecteur, définies par le vecteur $X_m = (x_m, y_m, z_m, \psi_x, \psi_y, \psi_z)$. A la plate-forme mobile (fig. A.3) correspond le vecteur opérationnel $X_b = (x_b, y_b, \theta)$ où θ est l'orientation de la plate-forme dans le référentiel fixe.

Les **coordonnées généralisées** traduisent la **configuration** du robot, elles sont liées aux actionneurs. On les définit dans le vecteur coordonnées généralisées $q = (q_1, \dots, q_{n_g})$ où n_g est la dimension de l'espace généralisé. Pour le bras manipulateur, on a $(\phi_1, \phi_2, \dots, \phi_{n_m})$ où n_m correspond au nombre d'articulations. Pour la base mobile, $q_b = (\phi_d, \phi_g)$ où ϕ_d et ϕ_g expriment les rotations des roues arrière droite et gauche respectivement, dans le cas où la plate-forme est non-holonome et son déplacement peut être complètement défini par les vitesses des roues arrière (la plupart des cas étudiés).

On définit le **degré de redondance** du robot comme la différence de dimension entre l'espace opérationnel et l'espace généralisé, soit $d = n_g - n_o$. La redondance d'un système traduit le fait qu'à une situation désirée donnée correspondent une infinité de configurations possibles.

On introduit la notion de contraintes et de critères de la manière suivante : une **contrainte** est une restriction physique qui peut être due aux limites mécaniques du système ou à l'environnement. Elle comprend une marge de sécurité fixée par l'utilisateur. Le **critère** est une expression mathématique qui va permettre au système de respecter une ou plusieurs contraintes. Une contrainte supplémentaire pourra bien sûr être fixée par l'utilisateur, de manière par exemple à déterminer une configuration *considérée* comme avantageuse au cours d'un déplacement libre.

A.2.3 Solutions orientées tâche

Dans la réalisation d'une tâche quelconque de manipulation mobile se pose le problème de l'inversion du modèle cinématique d'un système redondant. Le choix du ou des critères qui permettront de définir une configuration acceptable par rapport aux contraintes de réalisation de la tâche est d'autant plus difficile que la tâche se complexifie. Cette partie traite essentiellement de l'inversion du modèle cinématique et des critères qui lui sont ajoutés pour déterminer une configuration plus "confortable" du point de vue de la réalisation de la tâche.

Inversion des modèles

Si la séparation en deux sous-systèmes allège les modèles à traiter, cela ne résout pas complètement le problème de l'inversion de la jacobienne. Deux approches pour ce faire : la première consiste à rendre la matrice jacobienne carrée par l'ajout d'équations (autant que le degré de redondance) de manière à pouvoir l'inverser directement. Le modèle cinématique s'écrit alors sous la forme

$$\begin{bmatrix} J_b(q) & 0 \\ \frac{\partial x}{\partial q_b} & J_m(q) \\ & J_a(q) \end{bmatrix} \begin{bmatrix} \dot{q}_b \\ \dot{q}_m \end{bmatrix} = \begin{bmatrix} 0 \\ \dot{X}_e \\ \dot{w}_a \end{bmatrix} \quad \Rightarrow \quad J(q)\dot{q} = \dot{X}$$

où $J(q)$ est la jacobienne $n \times n$ du manipulateur mobile. L'inversion ne pose alors plus de problème et la matrice inverse de la jacobienne est définie tant que l'on se place en dehors des singularités (soit $l_1 \cos(\phi_1) + l_2 \cos(\phi_1 + \phi_2) \neq 0$). Il reste à déterminer les équations additionnelles $J_a(q)\dot{q} = \dot{w}_a$ les mieux adaptées à la tâche à réaliser. Par exemple imposer les vitesses généralisées sur le bras et la base [Seraji95] (celles de rotation de la plate-forme ($\dot{\theta} = \dot{\theta}_{dés}$) et du "coude" ($\dot{\phi}_2 = \dot{\phi}_{2dés}$)), ou seulement sur la base mobile (vitesses de déplacement et de rotation de la plate-forme imposées) ou seulement de la base mobile (par exemple l'épaule et le coude) permettra d'imposer la configuration du manipulateur mobile afin de contourner un obstacle. Les termes de la matrice \dot{w}_a peuvent être constants ou, comme c'est plus souvent le cas, dépendre du temps.

L'autre méthode consiste à utiliser une inverse généralisée $J^*(q)$ de la jacobienne, avec par définition $J(q)J^*(q)J(q) = J(q)$. Celle-ci propose, par le système d'équation $\dot{q} = J^*(q)\dot{X}$ une configuration possible. Pour obtenir une solution plus adaptée à la tâche (éviter d'obstacle, positionnement particulier de la base par rapport au bras,...), il faut intégrer au système un terme prenant en compte les critères désirés. Le modèle cinématique inverse s'écrit alors

$$\dot{q} = J^*(q)\dot{X} + (J^*(q)J(q) - I)Z$$

où I est la matrice identité d'ordre n et Z est un vecteur arbitraire. Dans cette relation le premier terme ($J^*(q)\dot{X}$) concerne les mouvements de l'organe terminal -la situation- et le deuxième ($(J^*(q)J(q) - I)Z$) influence les mouvements internes -la configuration- du manipulateur mobile. Pour une modélisation en efforts, on a le système équivalent

$$\Gamma = J^t(q)F + (I - J^t(q)J^{*t}(q))\Gamma_0$$

où comme pour le modèle cinématique inverse en vitesses $J^t(q)F$ concerne les efforts appliqués à l'organe terminal et $(I - J^t(q)J^{*t}(q))\Gamma_0$ ceux appliqués au bras. Khatib introduit l'inverse généralisée

$$J^*(q) = A^{-1}(q)J^t(q)\Lambda(q)$$

avec $A(q)$ et $\Lambda(q)$ les matrices d'inertie exprimées dans l'espace généralisé et opérationnel respectivement ($E_{cin} = \frac{1}{2}\dot{q}^t A(q)\dot{q} = \frac{1}{2}\dot{x}^t \Lambda(x)\dot{x}$).

Critères de restriction

cette classe de critères n'est pas destinée à déterminer une configuration optimale mais à définir un ensemble de **configurations interdites ou pénalisantes**. L'espace atteignable par le bras seul est alors restreint et la base est plus fréquemment obligée d'intervenir pour atteindre un domaine de manipulation acceptable du point de vue de ces critères. On en distingue deux types : au niveau local (d'une articulation), et au niveau global (du manipulateur mobile).

Le premier type contient l'éloignement des butées articulaires afin de limiter la fatigue mécanique du robot ainsi que des singularités de manière à ce qu'il ne perde pas de son efficacité. Nassal [Nassal96], par exemple, utilise deux fonctions de coût pour déterminer la configuration du manipulateur mobile au cours d'un quelconque mouvement. La *fonction de base* prend en compte l'évitement des limites articulaires ainsi que l'éloignement des singularités de l'épaule, du coude et de la main. Celle-ci s'écrit

$$\begin{aligned} \mathcal{C}_{basic} &= \mathcal{C}_{limit} + \mathcal{C}_{sing,bras} + \mathcal{C}_{sing,main} \\ &= \sum_{i=1}^n \left(\frac{1}{\phi_i - \phi_i^{min}} + \frac{1}{\phi_i^{max} - \phi_i} \right) + \left(\frac{1}{d_1(\phi)} + \frac{1}{d_2(\phi)} \right) + \frac{1}{\phi_5 - \phi_{5,sing}} \end{aligned}$$

où $d_1(\phi)$ et $d_2(\phi)$ sont les distances du poignet aux singularités de l'épaule et du coude respectivement. Une deuxième fonction coût, *additionnelle*, minimise les couples statiques pour chaque articulation du bras. Elle met en jeu les rapports entre les couples à l'instant considéré et les couples maximum, et est minimisée par un gradient négatif. Cette dernière fonction permet de diminuer la consommation du bras ainsi que d'augmenter la valeur de sa charge limite admissible.

Les critères de restriction globale considèrent par exemple l'équilibre du manipulateur mobile. L'espace atteignable par le bras est réduit de manière à éviter le renversement du robot lorsque le moment lié aux mouvements ou à la charge du bras dépasse les capacités de la base mobile. Tous les critères mis en place pour prendre en compte la stabilité du système utilisent le moment résultant des forces statiques et dynamiques ramenées en un point, par exemple le centre de gravité ([Ohkawa98], [Rey97]). Cette résultante doit rester inférieure à un vecteur qui dépend des caractéristiques de la base (poids, géométrie...) et de l'environnement (on doit par exemple tenir compte de l'évolution du robot sur une pente).

Une autre méthode d'évaluation de la stabilité du système considère que le système de locomotion du manipulateur mobile doit être miniaturisé pour réduire son encombrement, ou que le poids du bras manipulateur dépasse parfois largement celui de la base mobile [Huang97][Huang94]. La redondance est alors utilisée pour trouver une position du bras qui stabilise l'ensemble, quels que soient le mouvement du robot ou sa charge, en considérant la possibilité de perturbations extérieures soudaines. Il introduit le ZMP (Zero Moment Point), point sur le sol pour lequel la somme de tous les moments des forces actives (statiques ou dynamiques) est nulle. Le critère implique que le ZMP reste le plus loin possible des bords du domaine de stabilité du manipulateur mobile :

$$\alpha = \frac{d}{r_0} \quad \begin{cases} 0 \leq \alpha \leq 1 & \text{stable} \\ \alpha < 0 & \text{instable} \end{cases}$$

où α est le degré de stabilité du système, d est la distance du ZMP au bord du domaine de stabilité le plus proche et r_0 est la distance des points de stabilité maximale au bord du même domaine. Le domaine de stabilité est déterminé, à nouveau, par les caractéristiques de la base mobile, celles du bras (dont sa charge) et de l'environnement.

Les *critères de restriction* sont en général intégrés dans une fonction coût qui dépend d'autres critères qui permettront de déterminer la configuration finale du robot. Ces contraintes mécaniques peuvent ne pas être imposées explicitement, voir par exemple dans le paragraphe suivant la manipulabilité cinématique qui fait tendre "naturellement" les angles des articulations vers une valeur moyenne entre les deux extrémités, la singularité et la butée articulaire, sans faire apparaître leur valeur. Par contre, la restriction du domaine atteignable par le bras seul trouve son utilité par exemple lorsqu'un obstacle intervient et force une configuration limite.

Critères de configuration préférentielle

La configuration optimale du manipulateur mobile est celle qui lui permet une dextérité maximale à tout instant. Cette configuration n'est pas unique puisqu'elle dépend du critère considéré pour la déterminer d'une part, et d'autre part des contraintes de l'environnement ainsi que de la tâche à effectuer. Contrairement aux critères de restriction qui interdisent certaines configurations au robot, les critères de configuration vont faire tendre les coordonnées généralisées du manipulateur mobile vers une configuration, ou vers leur valeur optimale, à tout moment au cours d'un déplacement non contraint.

Huang étend la méthode du ZMP pour déterminer une configuration, en restreignant son degré de stabilité à 1. Cela revient à imposer à l'effecteur de rester sur la ligne de points équidistants des bords du domaine de stabilité.

La **manipulabilité cinématique** est la contrainte la plus utilisée pour positionner au mieux la base dans l'assistance permanente à la manipulation. Elle traduit une position particulièrement "confortable" pour le bras, dans le sens où tous les mouvements lui sont autorisés lorsqu'elle est maximale. Physiquement les articulations sont en angles droit, ce qui éloigne le bras manipulateur de ses singularités (angle nul) et de ses limites articulaires (en général elles se rapprochent de l'angle plat). Il y a plusieurs manières d'exprimer la manipulabilité cinématique, la plus courante étant par l'*indice de manipulabilité* ω :

$$\omega = \sqrt{|J(q)J^t(q)|},$$

lequel, lorsque proche de 1, éloigne au maximum les coordonnées généralisées du bras des positions singulières. Yamamoto a utilisé ce critère pour déterminer la configuration "préférée" du manipulateur mobile lors de ses déplacements ou lorsqu'il est tiré par l'effecteur par un humain. L'indice de manipulabilité permet alors d'influencer les mouvements de la base, de manière à ce que le bras soit prêt à manipuler à tout moment. Lors de la réalisation d'une tâche quelconque, un indice de manipulabilité constamment maximal suppose bien entendu que la base soit en constant déplacement et apporte une assistance permanente à la manipulation.

Dans le même objectif d'optimisation de la manipulabilité, Fischer et al. utilisent trois critères indépendants basés sur l'évitement de positions désagréables par l'humain pendant une tâche de manipulation. Ils optimisent l'*angle du coude*, tel que la distance entre le poignet et le véhicule tendent vers k_1 , l'*angle de l'épaule* (entre le véhicule et le bras) doit tendre vers k_2 et l'*angle du poignet* (entre la main et l'avant-bras) doit tendre vers k_3 . Ces trois critères sont exprimés par la fonction $k = f(\phi)$. Le manipulateur mobile part en configuration optimale, puis la base se positionne de telle manière que la valeur de $\Delta k = k_d - k$ soit minimale, k_d étant la configuration finale désirée.

Khatib utilise une fonction potentielle $V_c(q)$ dont le gradient, intégré dans le modèle cinématique inverse, permettra d'effectuer les modifications nécessaires avec l'utilisation d'une inverse généralisée pour obtenir les mouvements désirés.

$$V_c(q) = k_c \sum_i (q_i - \frac{\bar{q}_i + \underline{q}_i}{2})^2$$

où \bar{q}_i et \underline{q}_i sont les limites articulaires respectivement supérieure et inférieure de la liaison i . Le gradient de cette fonction fera donc tendre les angles de chaque articulation vers le milieu de ces deux limites.

Critère de configuration forcée

Ces critères permettent de forcer la configuration du manipulateur mobile de manière à prendre en compte les contraintes dues à l'environnement : évitement d'obstacle, que ce soit un obstacle ponctuel gênant pour manipuler, ou pour un suivi de mur. Forcer la configuration peut présenter certains risques, par exemple le renversement des robots ou accélérer l'usure par une utilisation extrême des capacités mécaniques. Les critères restrictifs trouvent donc leur utilité dans la prévention de ces situations.

Le suivi de mur est la tâche la plus fréquemment utilisée pour l'illustration de tels critères, ou plus exactement l'écriture sur un tableau, qui permet de réaliser en parallèle une tâche spécifique pour le bras et une pour la base mobile. La base est donc souvent dotée d'une trajectoire propre (calculée par un module d'évitement d'obstacle par exemple) et le bras s'adapte en fonction de la position de la base et de la trajectoire désirée de l'effecteur. Lorsque la configuration est forcée en fonction de la situation de l'organe terminal et de celle de l'obstacle, le critère correspondant apparaît par l'intermédiaire de la fonction coût, voir par exemple Seraji [Seraji95] qui optimise une fonction tenant compte des limitations des vitesses articulaires, de la contrainte de non-holonomie, de la priorité sur les mouvements de l'organe terminal et enfin des tâches additionnelles qui définissent entre autres les obstacles.

Un critère géométrique peut être déterminé facilement [Desai97], et encore une fois son gradient est utilisé. Ce critère inscrit la base (S_A) dans un cercle (de rayon r_A) qui ne doit pas entrer en contact avec l'obstacle (S_B). La fonction coût est définie par l'inégalité

$$\epsilon - d(S_A, S_B) \leq 0 \quad \text{avec} \quad d(S_A, S_B) = \sqrt{(x - x_0)^2 + (y - y_0)^2} - r_A$$

où ϵ est la distance minimale désirée à l'obstacle, (x, y) et (x_0, y_0) sont les coordonnées respectivement du robot et de l'obstacle dans le repère absolu.

A.2.4 Conclusion

La manipulation mobile ouvre beaucoup de perspectives du fait de la souplesse du système. C'est cette même souplesse qui pose problème lorsqu'il faut déterminer une

configuration judicieuse pour réaliser une tâche, et les solutions pour y parvenir sont peu nombreuses : on peut fixer les trajectoires que l'on sait adaptées à la tâche pour un robot (le bras ou la base) et l'autre robot s'adapte, ou déduire la configuration du manipulateur mobile de la situation à atteindre et de l'environnement. Pour cette dernière solution, la définition de contraintes et la mise en place des critères correspondants est nécessaire, qu'ils soient fixés par l'utilisateur ou calculés par le système. Ces critères apparaissent le plus souvent dans le modèle cinématique inverse sous la forme d'un vecteur qui est le gradient d'une fonction coût optimisée, et qui s'écrit sous la forme $g(q_i) \leq 0$. Un critère tel que l'indice de manipulabilité permet d'obtenir directement une relation entre les vitesses articulaires, mais il présente l'inconvénient de ne pas être homogène et donc difficilement interprétable.

Annexe B

Représentations mathématiques des chaînes cinématiques

B.1 Matrices de passage homogènes

La modélisation systématique et automatique des robots manipulateurs exige une méthode pratique et simple pour la description de leur morphologie. Les méthodes les plus usuelles sont décrites dans cette annexe. Elles utilisent la notion de matrices de passage homogènes [4x4], notées \mathbf{T}_i^j , pour décrire la transformation géométrique de \mathcal{R}_i à \mathcal{R}_j :

$$\mathbf{T}_i^j = \left(\begin{array}{ccc|c} & & & \\ & R_i^j & & t_i^j \\ & & & \\ \hline 0 & 0 & 0 & 1 \end{array} \right)$$

où t_i^j est un vecteur [3x1] qui décrit la translation de l'origine de \mathcal{R}_i à l'origine de \mathcal{R}_j . R_i^j est une matrice [3x3], aussi appelée matrice des cosinus directeurs, qui décrit la rotation entre \mathcal{R}_i et \mathcal{R}_j . Cette matrice contient 9 termes, r_{ij} , qui s'expriment en fonction de la convention choisie, reliés entre eux par 6 équations de contraintes :

$$\begin{aligned} r_{11} * r_{21} + r_{12} * r_{22} + r_{13} * r_{23} &= 0 \\ r_{11} * r_{31} + r_{12} * r_{32} + r_{13} * r_{33} &= 0 \\ r_{21} * r_{31} + r_{22} * r_{32} + r_{23} * r_{33} &= 0 \\ r_{11}^2 + r_{12}^2 + r_{13}^2 &= 0 \\ r_{21}^2 + r_{22}^2 + r_{23}^2 &= 0 \\ r_{31}^2 + r_{32}^2 + r_{33}^2 &= 0 \end{aligned}$$

La matrice de passage homogène pour traduire la transformation géométrique de \mathcal{R}_α à \mathcal{R}_β s'obtient en effectuant le produit des matrices homogènes associées aux transfor-

mations intermédiaires :

$$\mathbf{T}_\alpha^\beta = \mathbf{T}_\alpha^{\alpha+1} * \dots * \mathbf{T}_{\beta-1}^\beta = \prod_{k=\alpha+1}^{k=\beta} \mathbf{T}_{k-1}^k$$

B.2 Convention d'Euler

Elle permet de décrire la rotation du repère \mathcal{R}_j par rapport au repère \mathcal{R}_i par la composition de trois rotations successives d'angles (ψ, θ, ϕ) associés respectivement aux vecteurs $(\vec{Z}, \vec{X}', \vec{Z}'')$. La matrice de rotation résultante s'écrit alors :

$$\begin{aligned} R_i^j &= \begin{pmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & \sin \theta \\ 0 & -\sin \theta & \cos \theta \end{pmatrix} * \begin{pmatrix} \cos \phi & \sin \phi & 0 \\ -\sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} c_\psi c_\phi - s_\psi c_\theta s_\phi & s_\psi c_\phi + c_\psi c_\theta s_\phi & s_\theta s_\phi \\ -c_\psi s_\phi - s_\psi c_\theta c_\phi & -s_\psi s_\phi + c_\psi c_\theta c_\phi & s_\theta c_\phi \\ s_\psi s_\theta & -c_\psi s_\theta & c_\theta \end{pmatrix} \end{aligned}$$

où $c_\alpha = \cos \alpha$ et $s_\alpha = \sin \alpha$. Cette représentation peut caractériser l'orientation de l'organe terminal de la chaîne cinématique dans le repère de référence. Les angles correspondants s'obtiennent alors directement, dans le cas où $\sin \theta \neq 0$ [Craig86] :

$$\begin{aligned} \psi &= \text{Atan2}(r_{23}, r_{13}) \\ \theta &= \text{Atan2}(-\sqrt{r_{13}^2 + r_{23}^2}, r_{33}) \\ \phi &= \text{Atan2}(r_{32}, r_{31}) \end{aligned}$$

où r_{ij} correspond au terme de la $i^{\text{ème}}$ ligne et $j^{\text{ème}}$ colonne de la matrice de rotation.

B.3 Paramétrage de Denavit-Hartenberg

La convention introduite par [Denavit55] introduit quatre paramètres qui caractérisent totalement la transformation qui fait passer d'une articulation à la suivante dans une chaîne cinématique. Ces paramètres sont définis de la manière suivante (fig. B.1) :

$a_i \in \mathbb{R}^+$	la distance entre	A_{i-1} et A_i	mesuré selon	\vec{X}_i
$\alpha_i \in \mathbb{R}$	l'angle entre	\vec{Z}_{i-1} et \vec{Z}_i	mesuré autour de	\vec{X}_i
$d_i \in \mathbb{R}$	la distance entre	A_{i-1} et A_i	mesuré selon	\vec{Z}_{i-1}
$\theta_i \in \mathbb{R}$	l'angle entre	\vec{X}_{i-1} et \vec{X}_i	mesuré autour de	\vec{Z}_{i-1}

La matrice homogène de transformation pour passer de l'articulation $i-1$ à l'articulation i aura alors toujours la forme suivante :

$$\mathbf{T}_{i-1}^i = \left(\begin{array}{ccc|c} R_{i-1}^i & & & t_{i-1}^i \\ \hline 0 & 0 & 0 & 1 \end{array} \right) = \left(\begin{array}{ccc|c} \cos \theta_i & -\cos \alpha_i \sin \theta_i & \sin \alpha_i \sin \theta_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \alpha_i \cos \theta_i & -\sin \alpha_i \cos \theta_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ \hline 0 & 0 & 0 & 1 \end{array} \right)$$

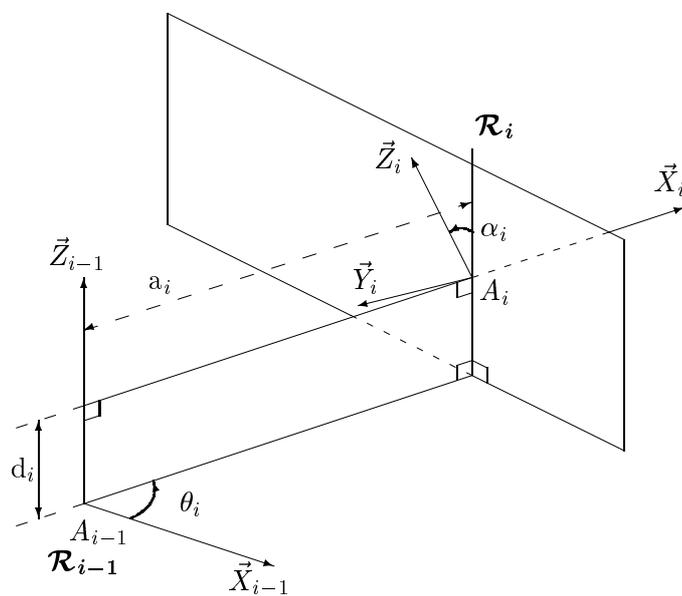


FIG. B.1: Convention de Denavit-Hartenberg.

Annexe C

Caractéristiques dynamiques de RoMAin et du PUMA 560

C.1 Modèle dynamique de ROMAIN

Cette plate-forme mobile a été conçue et réalisée par Pascal RUAUX dans le cadre d'une thèse de doctorat [Ruau98]. Le modèle dynamique correspondant pour la simulation a été validé par des résultats comportementaux comparables à ceux de la plate-forme réelle.

C.1.1 Equations de contraintes

Les contraintes de non-holonomie du véhicule traduisent l'impossibilité pour celui-ci de se déplacer dans toutes les directions (équations de contraintes non intégrables). Le modèle dynamique de la plate-forme mobile est établi sous les hypothèses suivantes :

- *Hypothèse 1* : Le contact roue-sol est un contact ponctuel
- *Hypothèse 2* : Le roulement de chaque roue s'effectue sans glissement
- *Hypothèse 3* : Le système évolue sur un sol plan horizontal

Les vitesses des roues arrières droite (d) et gauche (g) sont les suivantes :

$$\begin{aligned}\vec{v}_g &= (\dot{x}_G \cos \theta + \dot{y}_G \sin \theta - b\dot{\theta})\vec{x}_b + (-\dot{x}_G \sin \theta + \dot{y}_G \cos \theta - L_r\dot{\theta})\vec{y}_b \\ \vec{v}_d &= (\dot{x}_G \cos \theta + \dot{y}_G \sin \theta + b\dot{\theta})\vec{x}_b + (-\dot{x}_G \sin \theta + \dot{y}_G \cos \theta - L_r\dot{\theta})\vec{y}_b\end{aligned}$$

La composante de la vitesse est nulle suivant \vec{y}_b , nous obtenons donc la relation

$$-\dot{x}_G \sin \theta + \dot{y}_G \cos \theta - L_r\dot{\theta} = 0$$

De plus, les conditions de roulement sans glissement permettent d'établir le lien entre la vitesse de rotation des roues et le déplacement du centre de gravité de la plate-forme :

$$\begin{aligned}\dot{x}_G \cos \theta + \dot{y}_G \sin \theta - b\dot{\theta} &= r\dot{\phi}_g \\ \dot{x}_G \cos \theta + \dot{y}_G \sin \theta + b\dot{\theta} &= r\dot{\phi}_d\end{aligned}$$

Ces deux dernières équations permettent d'établir la relation entre la rotation de la plate-forme et les vitesses de rotations des deux roues :

$$\dot{\theta} = \frac{r}{2b}(\dot{\phi}_d - \dot{\phi}_g)$$

La manipulation de ces différentes relations permettent d'établir les deux équations de non-holonomie :

$$\begin{aligned} \dot{x}_G \cos \theta + \dot{y}_G \sin \theta - \frac{r}{2}(\dot{\phi}_d + \dot{\phi}_g) &= 0 \\ -\dot{x}_G \sin \theta + \dot{y}_G \cos \theta + \frac{rL_r}{2b}(\dot{\phi}_d - \dot{\phi}_g) &= 0 \end{aligned}$$

L'expression de la matrice des contraintes de non-holonomie est donc la suivante :

$$D(q)\dot{q} = [0]$$

où q est le vecteur des coordonnées généralisées :

$$q = \begin{pmatrix} x_G \\ y_G \\ \phi_g \\ \phi_d \end{pmatrix}$$

$$D(q) = \begin{bmatrix} -\sin \theta & \cos \theta & \frac{rL_r}{2b} & -\frac{rL_r}{2b} \\ \cos \theta & \sin \theta & -\frac{r}{2} & -\frac{r}{2} \end{bmatrix}$$

C.1.2 Modèle dynamique pour la simulation

Le formalisme de Lagrange est utilisé pour l'extraction du modèle dynamique pour la simulation. Il repose sur le calcul des énergies mises en jeu dans les déplacements du système. L'expression de l'énergie cinétique est la suivante :

$$T = \frac{1}{2}M_t(\dot{x}_G^2 + \dot{y}_G^2) + \frac{1}{2}I_{p33}\dot{\theta}^2 + \frac{1}{2}I_t(\dot{\phi}_g^2 + \dot{\phi}_d^2)$$

L'énergie potentielle K est nulle (hypothèse 3 précédente). L'application du formalisme de Lagrange permet donc d'écrire les équations dynamique du système sous la forme matricielle suivante :

$$M(q)\ddot{q} + B(q, \dot{q}) + G(q) = \tau' - D^t \lambda$$

avec :

- $M(q)$: matrice d'inertie
- $B(q, \dot{q}) = [0]$: matrice des effets de Coriolis et forces centrifuges
- $G(q) = [0]$: matrice des forces de gravité
- $\tau' = {}^t [0 \quad 0 \quad \tau_g \quad \tau_d]$: vecteur des couples moteurs

– $\lambda = {}^t [\lambda_1 \quad \lambda_2]$: vecteur des multiplicateur de lagrange.

et

$$M(q) = \begin{bmatrix} M_t & 0 & 0 & 0 \\ 0 & M_t & 0 & 0 \\ 0 & 0 & I_{p33}c^2 + I_{t22} & -I_{p33}c^2 \\ 0 & 0 & -I_{p33}c^2 & I_{p33}c^2 + I_{t22} \end{bmatrix} = \begin{bmatrix} 110 & 0 & 0 & 0 \\ 0 & 110 & 0 & 0 \\ 0 & 0 & 15,54 & 0 \\ 0 & 0 & 0 & 14,4 \cdot 10^{-3} \end{bmatrix}$$

L'élimination des multiplicateurs de Lagrange permet d'exprimer le modèle sous la forme

$$\tau = S^t(q)M(q)S(q)\dot{\nu} + S^t(q)M(q)\dot{S}(q)\nu$$

avec

$$\nu = \begin{bmatrix} \dot{\phi}_g \\ \dot{\phi}_d \end{bmatrix}$$

et

$$S(q) = \begin{bmatrix} \frac{rL_r}{2b} \sin \theta + \frac{r}{2} \cos \theta & -\frac{rL_r}{2b} \sin \theta + \frac{r}{2} \cos \theta \\ -\frac{rL_r}{2b} \cos \theta + \frac{r}{2} \sin \theta & \frac{rL_r}{2b} \cos \theta + \frac{r}{2} \sin \theta \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

C.2 Paramètres du PUMA 560

Le modèle dynamique du PUMA 560 est défini par ses paramètres DH (annexe C) et inertiels. Ceux-ci sont donnés dans le tableau C.1. Les correspondances de ces paramètres sont rappelées ci-après :

- Paramètres DH : α , A, θ , D. Leur définition précise est donnée en annexe C.
- Type d'articulation σ (0 : rotoïde, autre : prismatique)
- Masse du segment : m
- Position du centre de masse dans le repère lié au segment : $(r_x \quad r_y \quad r_z)^t$
- Tenseur d'inertie autour du centre de masse : $\begin{pmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{pmatrix}$
- Inertie de l'armature : J_m
- Rapport de réduction à l'articulation : G
- Frottements visqueux : B (actionneur)
- Forces de coulombs : T_{c+} et T_{c-} (respectivement rotation positive et négative).

	Axe 1	Axe 2	Axe 3	Axe 4	Axe 5	Axe 6
α	$\pi/2$	0	$-\pi/2$	$\pi/2$	$-\pi/2$	0
A	0	0.4318	0.0203	0	0	0
θ	0	0	0	0	0	0
D	0	0	0.15005	0.4318	0	0
σ	0	0	0	0	0	0
m	0	17.4	4.8	0.82	0.34	0.09
r_x	0	-0.3638	-0.0203	0	0	0
r_y	0	0.006	-0.0141	0.019	0	0
r_z	0	0.2275	0.070	0	0	0.032
I_{xx}	0	0.13	0.066	1.8e-3	0.3e-3	0.15e-3
I_{yy}	0.35	0.524	0.086	1.3e-3	0.4e-3	0.15e-3
I_{zz}	0	0.539	0.0125	1.8e-3	0.3e-3	0.04e-3
I_{xy}	0	0	0	0	0	0
I_{yz}	0	0	0	0	0	0
I_{xz}	0	0	0	0	0	0
J_m	200e-6	200e-6	200e-6	33e-6	33e-6	33^{-6}
G	-62.6111	107.815	-53.7063	76.0364	71.923	76.686
B	1.48e-3	0.817e-3	1.38e-3	71.2e-6	82.6e-6	36.7^{-6}
T_c+	0.395	0.126	0.132	11.2e-3	9.26e-3	3.96^{-3}
T_c-	-0.435	-0.071	-0.105	-16.9e-3	-14.5e-3	-10.5^{-3}

TAB. C.1: Caractéristiques dynamiques du PUMA 560

Annexe D

Sources des opérateurs évolutionnaires en langage C

D.1 Initialisation

La génération aléatoire des individus de la population initiale est choisie uniformément sur l'espace de recherche. La procédure est la suivante, pour des nombres binaires ou pour des réels :

```
for (i=0 ;i<nk ;i++) {                               /* Boucle sur la population */
  for (j=0 ;j<ns ;j++) {                             /* Boucle sur le chromosome */
    c = ran();
    BINAIRE[i][j] = c ;                               /* Chromosome binaire */
  }
  for (j=0 ;j<nt ;j++) {                             /* Boucle sur les tâches */
    for (k=0 ;k<nb ;k++) {                           /* Boucle sur le chromosome */
      r = drand48();
      REEL[j][i][k] = r ;                            /* Chromosome réel*/
    }
  }
}
```

D.2 Croisement

D.2.1 Croisement à 1 point de césure

```

for (i=0 ;i<k;i=i+2) {
c=drand48()*100.0 ;
if (c<pc) {
di=(rand() % ns) ;
for (j=0 ;j<ns;j++) {
if (j<di) {
Indiv[i][j]=Indiv2[i][j] ;
Indiv[i+1][j]=Indiv2[i+1][j] ;
}
if (j>=di) {
Indiv[i][j]=Indiv2[i+1][j] ;
Indiv[i+1][j]=Indiv2[i][j] ;
}
}
}
}

```

D.2.2 Croisement uniforme

```

for (i=0 ;i<k;i=i+2) {
c=drand48()*100.0 ;
if (c<pc) {
di=(rand() % ns) ;
for (j=0 ;j<ns;j++) {
s=gat()%2 ;
if (s==0) {
Indiv[i][j]=Indiv2[i][j] ;
Indiv[i+1][j]=Indiv2[i+1][j] ;
}
if (s==1) {
Indiv[i][j]=Indiv2[i+1][j] ;
Indiv[i+1][j]=Indiv2[i][j] ;
}
}
}
}
}

```

D.2.3 Reproduction des individus

```
for (i=0 ;i<k ;i=i+2) {
    for (j=0 ;j<ns ;j++) {
        Indiv[i][j]=Indiv2[i][j] ;
        Indiv[i+1][j]=Indiv2[i+1][j] ;
    }
}
```

D.3 Mutation

D.3.1 Mutation binaire

```
for (i=0 ;i<nk ;i++) {
    for (j=0 ;j<ns ;j++) {
        ra=(float)(drand48()*100.0) ;
        if (ra<pm)
            Indiv[i][j]=(Indiv[i][j]+1)%2 ;
    }
}
```

D.3.2 Mutation réelle

```
for (i=0 ;i<nk ;i++) {
    for (t=0 ;t<nt ;t++) {
        for (j=0 ;j<nb ;j++) {
            ra=(float)(drand48())-0.5 ;
            Indiv[t][i][j]=Indiv[t][i][j] + pm*ra ;
            if (Indiv[t][i][j]<0) Indiv[t][i][j]=0 ;
            if (Indiv[t][i][j]>1) Indiv[t][i][j]=1 ;
        }
    }
}
```