

**MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE
SCIENTIFIQUE**

**UNIVERSITÉ FERHAT ABBAS SÉTIF
UFAS (ALGERIE)**

THESE

Présentée à la faculté des sciences de l'Ingénieur

Département d'Electronique

Pour l'Obtention du Diplôme de

DOCTORAT D'ETAT

Option:Électronique

Par

MR: Hassam Abdelouahab

THEME :

**CALCUL DE TRAJECTOIRE ET CONTRÔLE DE DEPLACEMENT
D'UN ROBOT**

Soutenu le 15 Décembre 2007 devant la commission d'examen :

Pr. A. KHELLAF	Prof. Université de Sétif	Président
Pr. K. BENMAHAMMED	Prof. Université de Sétif	Rapporteur
Pr. N. BOUGUECHAL	Prof. Université de Batna	Examineur
Pr. R. ABDESSAMED	Prof. Université de Batna	Examineur
Pr. N. KHENFER	Prof. Université de S é tif	Examineur

Remerciements

Ce travail de thèse a été réalisé au sein du Laboratoire de Systèmes Intelligents de la Faculté des Sciences de l'Ingénieur de l'Université Ferhat Abbès de Sétif, en collaboration avec d'autres Laboratoires.

Qu'il me soit permis de remercier toutes les personnes qui, à des titres divers, ont participé de près ou de loin à l'élaboration de ce travail.

Ainsi, je voudrais exprimer ma profonde gratitude et ma sincère reconnaissance à Monsieur le Professeur **BENMAHAMMED KHIER** du Département d' Electronique et directeur de ma thèse, pour la confiance qu'il m'a toujours témoignée. J'ai tout particulièrement apprécié ses conseils et ses encouragements, surtout en fin de thèse. Je lui exprime ici toute mon admiration pour ses qualités scientifiques, son esprit et son dynamisme dont il a fait preuve pour diriger mon travail.

Sa collaboration m'a permis de progresser et de comprendre de ne jamais abandonner en recherche malgré toutes éventuelles difficultés.

J'adresse mes sincères remerciements à Madame le Professeur **ZAHIA GUESSOUM** du Laboratoire d'Informatique de Paris (LIP 6), qui m'a accueilli à maintes reprises dans son laboratoire pour m'orienter dans mes recherches. Elle m'a de la sorte beaucoup aidé dans la réalisation de ce travail.

Je suis heureux de lui faire-part de ma gratitude et de ma sincère reconnaissance.

Je remercie également Monsieur le Professeur **FAIZ BENAMAR** qui m'a reçu dans son Laboratoire de Robotique de Paris (LRP). Son orientation et ses remarques ont pleinement contribué au développement de mon travail.

Je remercie les membres du jury pour m'avoir fait l'honneur de juger ce travail :

- Monsieur **KHELLAF ABDELHAFID.**, Professeur au Département d'Electronique de l'Université Ferhat Abbès de Sétif que je remercie de l'attention qu'il a toujours manifestée à mon égard et qui m'a fait l'honneur de présider ce dit jury.

- Monsieur **BOUGUECHAL NOUREDINE**, Professeur au Département d'Electronique de l'Université de Batna, pour l'intérêt qu'il a bien voulu porter à ce travail, en m'honorant de l'examiner.

- Monsieur **ABDESSAMED RACHID**, Professeur au Département d'Electrotechnique de l'Université de Batna, d'avoir accepté, sans hésitation aucune et avec bonne foi, de participer au jury de cette thèse.

- Monsieur **NABIL KHENFER.**, Professeur au Département d'Electronique de l'Université Ferhat Abbès de Sétif à qui j'adresse vivement ma reconnaissance pour ses conseils et ses encouragements.

Enfin, je tiens à remercier particulièrement toute ma famille et tous mes amis pour leur soutien moral qui me fut d'une grande utilité.

MERCI INFINIMENT A VOUS TOUS !

Table des matières

Résumé	v
Liste des figures	vi
Liste des tableaux	ix
1. Introduction	1
1.1. Synthèse des travaux réalisés	3
1.2. Organisation de la thèse	6
1.3. Conclusion	8
2. La manipulation mobile	9
2.1 Introduction	9
2.2 Stabilité des manipulateurs mobiles	10
2.3 Les applications des manipulateurs mobiles	11
2.3.1. Le domaine terrestre	11
2.3.2.. Le domaine spatial	11
2.3.3.. Le domaine sous terrain	11
2.4 Les contraintes des manipulateurs mobiles	11
2.5 Commande des manipulateurs mobiles	12
2.6 Constitution et modélisation des robots manipulateurs	14
2.6.1 Définition d'un robot manipulateur	14
2.6.2 Constitution d'un robot manipulateur	14
2.6.2.1. Structure mécanique d'un robot manipulateur	16
2.6.2.2. Les actionneurs	20
2.6.2.3. Les capteurs	23
2.7. Modélisation des robots manipulateurs	26

2.7.1 Introduction.....	26
2.7.2. Modélisation géométrique	26
2.7.3. Modélisation cinématique.....	30
2.7.4. Modélisation dynamique.....	31
3. Les algorithmes génétiques	38
3.1. Introduction.....	38
3.2. Principe de fonctionnement des algorithmes génétiques	39
3.3. Ce qui rend les AGs différents des méthodes classiques ?	40
3.4. Formulation du problème d'optimisation	41
3.5. Les mécanismes d'un algorithme génétique standard.....	42
3.5.1. La sélection	42
3.5.2. Le croisement.....	43
3.5.3. La mutation	44
3.6. Notion du codage.....	45
3.7. Notion du décodage.....	45
3.8. Théorie fondamentale des AGs (notion de sheme).....	46
3.8.1. Effet de la sélection	47
3.8.2. Effet du croisement	47
3.8.3. Effet de la mutation	48
3.8.4. Effet combiné.....	48
3.9. Amélioration des techniques de base des AGs	49
3.9.1. Amélioration des opérateurs de reproduction.....	49
3.9.2. Mécanisme de changement d'échelle	51
3.9.3. Optimisation des paramètres de l'algorithme génétique	52
3.9.4. Introduction des contraintes.....	53
3.10. Conclusion.....	54
4. La logique floue	55
4.1. Histoire de logique floue	55
4.2. Théorie des ensembles flous	56
4.2.1. Concepts fondamentaux flous.....	56

4.2.2. Définition d'un ensemble flou.....	57
4.2.3. Caractéristiques d'un ensemble flou.....	57
4.2.4. Fonction d'appartenance d'un ensemble flou.....	58
4.3. Variables floues et linguistiques	60
4.4. Opération sur les ensembles flous	60
4.4.1. L'égalité.....	60
4.4.2. L'inclusion.....	60
4.4.3. L'intersection.....	60
4.4.4. L'union.....	61
4.4.5. La complémentation	61
4.5. Normes et co-normes triangulaires.....	62
4.5.1. Normes triangulaires	62
4.5.2. Co-normes triangulaires.....	62
4.6. Relations floues.....	63
4.6.1. Composition de relations floues	63
4.7. Dédution floue et raisonnement approximatif.....	63
4.8. Commande et réglage par la logique floue.....	64
4.9. Structures des systèmes a base de la logique floue.....	65
4.9.1. L'interface de fuzzification	66
4.9.2. La base de connaissances floues.....	66
4.9.3. L'engin d'inférence	66
4.9.4. L'interface de défuzzification.....	66
4.10. Conclusion.....	67
5. Planification de mouvement des robots manipulateurs	68
5.1. Introduction.....	68
5.2. Problématique de génération de mouvement.....	68
5.2.1. Trajectoires totalement spécifiées	69
5.2.2. Trajectoires partiellement spécifiées.....	69
5.2.3. Trajectoires non spécifiées.....	69
5.2.4. Génération de mouvement dans l'espace articulaire.....	71
5.2.5. Génération de mouvement dans l'espace opérationnel.....	72
5.3. Les approches de planification de mouvement.....	72

5.3.1. Les techniques classiques.....	72
5.3.2. Les techniques récentes.....	73
5.4. Conception d'un planificateur de mouvement par les AGs.....	73
5.4.1. Formulation du problème de planification de mouvement.....	74
5.4.2. Description de l'algorithme génétique proposé.....	75
5.5. Résultats de simulation.....	80
5.6. Conclusion.....	85
6. Contrôle de mouvement des robots manipulateurs	86
6.1. Introduction.....	86
6.2. Les approches de contrôle de mouvement.....	86
6.2.1. Approche connexioniste.....	87
6.2.2. Approche directe.....	87
6.3. Conception d'un contrôleur flou par les AGs.....	87
6.3.1. Description du contrôleur flou proposé.....	88
6.3.2. Définition des paramètres à optimiser.....	90
6.3.3. Codage des paramètres.....	91
6.3.4. Fonction d'objectif.....	92
6.3.5. Les opérateurs génétiques.....	93
6.4. Description et modélisation du robot manipulateur.....	93
6.4.1. Description du robot manipulateur.....	93
6.4.2. Modélisation géométrique du robot.....	94
6.4.2. Modélisation dynamique du robot.....	96
6.5. Résultats de simulation.....	99
6.6. Conclusion.....	108
Conclusion générale	109
Références bibliographiques	112

Résumé

Le travail présenté dans ce mémoire fait partie du domaine de la planification de mouvement en robotique. Nous envisageons l'élaboration d'une structure de contrôle capable de gérer des tâches de manipulation mobile. Ce type de tâches fait intervenir un organe de locomotion (plate-forme mobile) et un organe de travail (bras manipulateur). Il s'agit d'une structure de commande séparée du bras et de la plate-forme.

Dans cette étude, un système de planification et de contrôle de mouvement des robots manipulateurs est donc développé, le système proposé est divisé en deux blocs : le premier bloc est destiné à la planification de mouvement des robots manipulateurs, il utilise un algorithme génétique pour générer la trajectoire optimale permettant au robot d'éviter toute collision avec d'éventuels obstacles existant dans son espace de travail. Le deuxième bloc est un contrôleur flou optimisé par un algorithme génétique, il est chargé de contrôler en permanence le bon suivi des trajectoires générées par le planificateur de mouvement.

L'efficacité du système proposé est évaluée à travers la simulation d'un robot manipulateur à deux degrés de liberté.

De ce manipulateur mobile, nous avons traité que le « bras robotisé » en mode autonome. En effet, le contrôle du bras seul est suffisant pour illustrer les problèmes de supervision propres à notre architecture de contrôle à réaliser.

Liste des figures

2-1	Structure générale d'un robot manipulateur	15
2.2	Structure à chaîne simple ouverte	16
2.3	Structure à chaîne arborescente	17
2.4	Structure à chaîne complexe (fermée).....	17
2.5	Symboles des liaisons rotoïde et prismatique.....	18
2.6	Organisation d'une installation hydraulique	21
2.7	Evolution du couple en fonction de la vitesse. Commande d'inducteur.	22
2.8	Evolution du couple en fonction de la vitesse. Commande d'induit.....	22
2.9	Principe de fonctionnement d'un moteur pas à pas.....	23
2.10	Les principaux capteurs de position rencontrés en robotique.	24
2.11	Les différentes parties d'un capteur externe.	25
2.12	Transformation de translation pure.	27
2.13	Transformation de Rotation pure.	28
2.14	Représentation graphique de la méthode de D.H	29
2.15	Représentation graphique de 0r_i et ${}^i r_i$	33
3.1	Organigramme d'un AG standard.	40
3.2	Croisement simple.....	44
3.3	Opération de mutation	44
3.4	Opérateur de croisement en deux points	50
3.5	Opérateur de croisement aléatoire.....	51
3.6	Algorithme génétique adaptatif.....	53
4.1	Ensemble flou : (a) convexe, (b) non-convexe	58
4.2	Fonction triangulaire	59
4.3	Fonction trapézoïdale	59

4.4	Fonction gaussienne	59
4.5	Fonction en forme de cloche.....	59
4.6	Fonction sigmoïde.....	59
4.7	Structure de base d'un système flou.	65
5.1	Génération de la trajectoire totalement spécifiée.....	69
5.2	Génération de la trajectoire non spécifiée.	70
5.3	Génération de mouvement dans l'espace articulaire	71
5.4	Génération de mouvement dans l'espace opérationnel	71
5.5	Structure du planificateur de mouvement	74
5.6	Les positions des joints dans l'espace cartésien.....	74
5.7	Structure du chromosome à deux dimensions	76
5.8	Crossover analogique.	76
5.9	Mutation addition	77
5.10	Mutation soustraction	77
5.11	Organigramme du planificateur de mouvement	79
5.12	Structure du robot à 2 d.d.l.....	80
5.13	Structure du chromosome généré par l'AG	80
5.14	Evolution de la fonction d'objectif	81
5.15	Trajectoire optimale non contrainte	81
5.16	Séquence de variations des angles des joints.....	82
5.17	Trajectoire optimale contrainte avec un obstacle	82
5.18	Séquence de variations des angles des joints avec un obstacle.....	83
5.19	Trajectoire optimale contrainte avec deux obstacles	83
5.20	Séquence de variations des angles des joints avec deux obstacles.....	83
5.21	Trajectoire optimale contrainte avec changement d'un des deux obstacles	84
5.22	Séquence de variations des angles des joints après changement des obstacles	84
6.1	Structure du système de contrôle.	88
6.2	Etapas de conception d'un contrôleur flou.....	89
6.3	Les paramètres décrivant la fonction d'appartenance	90
6.4	Structure du robot à 2 d.d.l.	94

6.5	Placement des repères sur les joints.....	94
6.6	Trajectoire à poursuivre par l'organe terminal.....	100
6.7	Trajectoires à poursuivre par chacune des articulations	100
6.8	Evolution de la fonction d'objectif.....	101
6.9	Fonctions d'appartenance de la variable $Er1$	102
6.10	Fonctions d'appartenance de la variable $Der1$	102
6.11	Fonctions d'appartenance de la variable $Cp1$	103
6.12	Fonctions d'appartenance de la variable $Er2$	103
6.13	Fonctions d'appartenance de la variable $Der2$	104
6.14	Fonctions d'appartenance de la variable $Cp2$	104
6.15	Performances de poursuite dans l'espace cartésien.....	105
6.16	Performances de poursuite dans l'espace articulaire.....	105
6.17	Erreurs de poursuite dans l'espace articulaire.....	106
6.18	Performances de poursuite dans l'espace cartésien sous des conditions sévères.....	106
6.19	Performances de poursuite dans l'espace articulaire sous des conditions sévères.....	107
6.20	Erreurs de poursuite dans l'espace articulaire sous des conditions sévères.	107

Liste des tableaux

6.1	Codage de la base des règles floues	91
6.2	Codage binaire des règles floues.....	91
6.3	Codage des paramètres des fonctions d'appartenance	92
6.4	Paramètres mécaniques du robot	94
6.5	Les paramètres de D.H du robot PLANAR.....	95
6.6	Valeurs des paramètres optimisés par L'AG.....	101
6.7	Base de règles floues du premier contrôleur	104
6.8	Base de règles floues du deuxième contrôleur.	105

Chapitre 1

Introduction

En robotique mobile, la problématique n'est plus seulement l'aspect navigation ou manipulation, mais surtout l'étude des interactions entre le robot et l'environnement dans une tâche de travail. Ainsi pour élargir les champs d'action d'un bras manipulateur, il s'avère nécessaire de le placer sur une plate-forme mobile constituant ainsi un manipulateur mobile. L'espace de travail se voit alors croître permettant ainsi de réaliser des tâches de manipulation sur un espace de travail plus important. Un manipulateur mobile est donc composé d'une plate-forme mobile sur laquelle est installé un bras manipulateur. Ils présentent des avantages indéniables par rapport aux manipulateurs fixes. Ils permettent, par exemple, d'étendre l'espace de travail de l'organe terminal du manipulateur qui, de borné, devient infini. Ils ont deux particularités: ils sont fortement redondants et ils sont en général non-holonomes. La redondance est liée au fait que le nombre de degrés de mobilité du manipulateur est, en général, supérieur au degré de liberté de son organe terminal, la non-holonomie est due aux conditions de locomotion de la plate-forme lorsqu'il y a roulement sans glissement sur le sol. La redondance induit une infinité de mouvements possibles du manipulateur mobile, pour un mouvement désiré de son organe terminal, alors que la non-holonomie interdit certains mouvements à la plate-forme.

Ces systèmes ont suscité un intérêt considérable dans de nombreux domaines. Leur utilisation est envisageable dans le transport et la manipulation d'objets, la maintenance des centrales nucléaires, le déminage, l'exploration des océans, la manipulation chimique ou autres. Ces services robotiques nécessitent des systèmes performants avec une grande flexibilité. Seulement, on est souvent confronté à d'autres problèmes liés à la redondance, à l'interaction dynamique entre la plate-forme mobile et le bras manipulateur. Par conséquent, la modélisation, la planification et le contrôle des manipulateurs mobiles demeurent des axes de recherches de nombreux laboratoires.

De ces manipulateurs mobiles, nous avons traité que le « bras robotisé » en mode autonome. En effet, le contrôle du bras seul est suffisant pour illustrer les problèmes de supervision propres à notre architecture de contrôle à réaliser. Ultérieurement, il sera possible de considérer le robot dans son intégralité en développant les contrôles du bras en mode autonome (travaux déjà réalisés et exposés dans notre thèse) ainsi que ceux de la plate forme mobile (travaux initiés) et les deux couplés (travaux en cours de réalisation). Mais dans un premier temps, seul le contrôle du bras manipulateur en mode autonome a été abordé .

L'exploitation de ces systèmes (robots manipulateurs) dans des opérations industrielles telles que la manipulation et l'assemblage pour augmenter la productivité, uniformiser la qualité des produits finis et encore diminuer le coût de production exige la présence des outils très efficaces pour leur contrôle.

Contrôler un robot manipulateur pour exécuter une tâche bien précise consiste à résoudre deux problèmes :

Le premier est la planification de la trajectoire que le robot doit poursuivre pour réaliser la tâche désirée dans un temps minimal et sans collision avec les obstacles existant dans l'espace de travail du robot.

Le deuxième problème est le contrôle de mouvement du robot de façon à ce que ce dernier puisse suivre fidèlement la trajectoire générée par le planificateur de mouvement sans dépassements ni oscillations.

Le problème de planification de mouvement doit être tout d'abord transformé en un problème d'optimisation de trajectoires sous la contrainte qu'aucune partie du robot n'entre en collision avec les obstacles existant dans son espace de travail, ensuite on doit utiliser une méthode d'optimisation parmi celles existantes. Du fait de la complexité du problème à résoudre, la seule solution est d'utiliser les algorithmes génétiques à cause de leur puissance importante d'optimisation.

En ce qui concerne le problème de contrôle de mouvement, nous avons jugé utile d'utiliser un contrôleur flou optimisé par un algorithme génétique. L'utilisation du contrôle flou a éliminé le besoin du modèle mathématique exact représentant le système à contrôler (Robot manipulateur) et l'utilisation de l'AG a éliminé le besoin d'un expert humain dans la phase de conception de tel contrôleur.

Dans ce travail, notre souci principal était d'ajouter une nouvelle dimension d'intelligence artificielle à la robotique, cette dimension est représentée par la génération automatique de trajectoires et l'extraction automatique de connaissances nécessaires à la conception des contrôleurs flous efficaces et robustes.

Le travail présenté dans ce mémoire fait partie du domaine de la planification de mouvement en robotique. Les recherches en la matière sont très actives depuis plusieurs années .

Les robots concernés par ces travaux de recherche sont des robots de type bras articulé, plate-forme mobile, et le problème traité est celui de la détermination automatique de mouvements pour amener un robot manipulateur d'une position initiale à une position finale. Les mouvements en question doivent satisfaire un ensemble de contraintes, et éventuellement optimiser certains critères.

Notre travail a donc pour objectif de définir une structure de contrôle capable de gérer des tâches de manipulation mobile. Ce type de tâche fait intervenir un organe de locomotion en l'occurrence la plate-forme mobile et un bras manipulateur constituant l'organe de travail. Cette structure de contrôle doit donc gérer les comportements futures des deux organes pour réaliser une tâche robotique composée à la fois de phase de navigation et de travail en intégrant en plus les interactions avec l'environnement. Dans cette thèse, nous n'envisageons que le cas d'une commande séparée du bras manipulateur et de la plate-forme.

1.1 : Synthèse des travaux réalisés :

Remarquablement, les travaux effectués dans le domaine de la manipulation mobile ont eu une grande attention durant les dernières quinze années à cause de leurs applications et de leurs problèmes analytiques liés à leurs commandes et à la planification de trajectoires. Un bras manipulateur mobile est composé d'un bras manipulateur fixé sur une plate-forme mobile. Les raisons de concevoir de tels systèmes sont de pouvoir augmenter l'espace de travail des bras manipulateurs et leur permettre d'atteindre toutes les positions désirées dans la configuration la plus optimale possible. Mais, dès lors qu'on place le bras manipulateur sur la plate-forme mobile que le problème de redondance apparaît suite aux degrés supplémentaires apportés par la plate-forme mobile ; mais aussi de la contrainte non-holonome imposée par la nature des roues de cette dernière. Seulement, ceci ne peut être qu'un avantage lorsqu'on sait qu'un système redondant peut éviter les obstacles et les configurations singulières et reste loin

des saturations articulaires. Il se trouve cependant que la présence de la contrainte non-holonome diminue la flexibilité des degrés de liberté disponibles, chose qui peut affecter la génération de trajectoires et réduit l'espace de travail. Dans cette partie, nous présentons brièvement les publications et les travaux rencontrés dans la littérature concernant la manipulation mobile et qui ont suscité un grand intérêt dans la communauté scientifique. Son but doit contribuer à designer les problèmes d'un manipulateur mobile liés à la conception de la tâche et à l'élaboration des méthodes de contrôle coordonné de l'ensemble.

Le problème de planification globale de la tâche peut être formulé comme un problème d'optimisation dans lequel les variables de déplacement de la plate-forme mobile sont séparées des variables articulaires dans la fonction coût. Le problème numérique résultant est non-linéaire présentant des régions non-convexes et non-connectés dans l'espace de décision, et qui peut être résolu par l'utilisation des algorithmes génétiques [1]. D'autre part, pour un manipulateur mobile qui effectue une séquence de tâches, la configuration finale de chaque tâche devient la configuration initiale des tâches qui succèdent. Ce concept suggéré par Pin et al.[2], utilise la redondance pour optimiser la configuration du robot entre deux tâches que le robot est susceptible d'effectuer. Ils s'intéressent au problème de l'optimisation locale dans l'espace des coordonnées généralisées par un critère global regroupant la minimisation des couples, l'évitement de collision et l'indice de manipulabilité qui permet d'éviter les configurations singulières. Les auteurs ont observé que cette méthode était plus efficace que celles utilisant un seul critère. Dans une autre publication, Pin et al [3] proposent une méthode qu'ils ont appelée *FSP (Full Space Parametrisation)* pour résoudre d'une façon optimale le problème de cinématique inverse d'un système redondant en présence de contraintes. Cette méthode fût ensuite améliorée par le travail de Lee *et al.*[4], qui proposent de résoudre le problème de planification de trajectoire en deux étapes. Dans une première phase, ils obtiennent les équations cinématiques différentielles du manipulateur mobile, puis ils cherchent l'ensemble des configurations qui minimise une certaine fonction objective en présence de contraintes. Des simulations ont été faites sur une plate-forme holonome et étendues à une plate-forme non-holonome. Seraji [5], résout la redondance par la formulation des tâches additionnelles qui permet d'obtenir un système d'équations décrivant le système mécanique non redondant. Une fonction de coût est alors déterminée à partir de ces tâches afin de calculer les variables articulaires. Les degrés de liberté de la base mobile et du bras

manipulateur contribuent à l'exécution des tâches de base et des tâches additionnelles. Une pondération basée sur les vitesses articulaires maximales et admissibles sur chaque axe est effectuée de façon à tenir compte de la dynamique de chaque sous-système. Parmi les techniques de planification de trajectoires en présence d'obstacles, on peut noter la méthode proposée par Quinlan [6] pour les plate-formes mobiles et qui fut appelée bande élastique (*Elastic band*) et qui n'est rien d'autre qu'une trajectoire déformable dans l'espace libre des configurations de la plate-forme mobile. On modélise son volume accessible, sans collision, par des bulles dans l'espace opérationnel, dont la taille est plus ou moins grande selon que l'on s'approche des obstacles. Brock a étendu ce concept aux manipulateurs mobiles et l'a appelé (*Elastic strips*) [7]. Il s'agit d'une approche qui permet de déformer une trajectoire donnée pour prendre en compte d'éventuels obstacles en temps réel et permet un meilleur contrôle de la posture du robot. Cette approche peut échouer, cependant, quand les couples résultant de l'application des couples articulaires dans l'espace nul résultent en un mouvement insuffisant pour accomplir le comportement désiré. C'est le cas par exemple dans les évitements d'obstacles où la tâche doit être suspendue en réservant tous les degrés de liberté pour prévenir une éventuelle collision. La tâche peut être reprise en cas de possibilité. Brock propose dans ce cas-ci des critères pour déterminer quand une telle transition est exigée [8].

La méthode des moindres carrés amortie pour minimiser la fonction de coût scalaire est utilisée. Les travaux de Nassal [9], porte sur le développement d'une commande dite transparente. Son approche est destinée au contrôle et à la coordination d'une base mobile supportant deux bras manipulateurs. Le principe consiste en ce que le déplacement de la plate-forme n'affecte pas la vitesse de l'organe terminal dans l'espace de travail. Autrement dit, la plate-forme est transparente pour l'organe terminal. La seconde tâche consiste à maintenir le manipulateur, par la base mobile, dans une configuration optimale au sens de la manipulabilité. Parallèlement, nous pouvons déceler des travaux remarquables dans la planification de la tâche et le contrôle des manipulateurs mobiles en considérant leurs modèles dynamiques. Dans l'article de Yamamoto et Yun [10], on trouve un algorithme de contrôle d'une plate-forme non-holonome à roues. Les déplacements mesurés des articulations du bras manipulateur sont pris comme entrées pour la planification du mouvement et commandent la plate-forme de telle sorte que l'organe terminal soit placé dans une zone de configuration préférentielle déterminée à partir de l'indice de

manipulabilité [11]. L'algorithme de coordination entraîne donc un mouvement du véhicule en fonction de la réponse aux mouvements du manipulateur, tout en tenant compte des contraintes non-holonomes de la base. Le calcul des maxima de l'indice de manipulabilité, permet de définir des domaines privilégiés de l'espace de travail. A chaque domaine, est associé un point unique auquel l'algorithme de poursuite essaye de le confondre avec le point final de l'effecteur. Afin de concevoir un contrôleur découplé pour un système non-holonome, Tahboub [12] propose une commande robuste pour contrôler l'organe terminal d'un bras manipulateur mobile en utilisant un observateur afin d'estimer la partie non-linéaire. Il modélise le mouvement de la plate-forme comme étant une perturbation dans son système d'équations puis utilise les concepts du contrôle linéaire pour une poursuite optimale. Dans les travaux de Khatib [13] [14], on voit qu'il considère le manipulateur mobile comme une combinaison d'un système macro/mini. Leur but est de traiter l'ensemble comme un seul système redondant. Avec cette stratégie la «macro » est la base mobile considérée ici comme un système ayant une réponse dynamique lente mais qui offre un espace opérationnel infini. La «mini » est le bras manipulateur considéré comme rapide et précis mais qui bute facilement à cause de la saturation de ses articulations. Leur méthode fût appliquée à un bras manipulateur type PUMA 560, monté sur une plate-forme holonome type Nomad 200.

1.2 Organisation de la thèse :

La technique et les recherches dans le domaine de la manipulation mobile ayant Considérablement progressés, les systèmes continuent à gagner en versatilité et en applicabilité.

Dans cette thèse, nous nous proposons à contribuer dans la gestion de la tâche d'un manipulateur mobile. Notre but est de contrôler l'effecteur du bras manipulateur en temps réel, alors que le bras manipulateur se déplaçant en conséquence afin de garantir la poursuite de la trajectoire désirée.

Partant de l'idée que le bras manipulateur mobile est une structure mécanique à deux composantes, nous avons choisi une commande séparée pour chacune d'elle. En considérant que le bras manipulateur possède une réponse rapide et précise, ce qui nous a conduit à contrôler le robot manipulateur pour exécuter une tâche bien précise. Contrôler le robot manipulateur consiste donc à résoudre les deux problèmes de la planification de trajectoire et le

contrôle de mouvement du robot .

Notre contribution consiste à effectuer une étude profonde des manipulateurs mobiles et d'ajouter une nouvelle dimension d'intelligence artificielle à la robotique, cette dimension est représentée par la génération automatique de trajectoires et l'extraction automatique de connaissances nécessaires à la conception des contrôleurs flous efficaces et robustes. Le but d'utiliser la logique floue est justifié par le fait que l'environnement dans lequel évolue la structure mécanique est peu connu donc, non modélisable, les variables mesurées sont variables et peu précises. Dans ce sens, les contrôleurs flous affichent un bon degré de robustesse face à ces variabilités et incertitudes. Ainsi, nous avons organisé cette thèse en six chapitres.

Dans ce *Chapitre un* nous situons notre problème à travers une étude bibliographique en donnant une synthèse des travaux réalisés.

Dans le *Chapitre deux*, nous nous consacrons à donner les principes fondamentaux qui entrent dans la composition des manipulateurs mobiles en présentant la plate-forme mobile ainsi que la constitution et la modélisation des bras manipulateurs.

Nous abordons dans le *Chapitre trois*, une description générale des principes de fonctionnement des algorithmes génétiques, les mécanismes de l'AG standard et les améliorations de l'AG standard y sont exposés .

Le *Chapitre quatre* est consacré à l'étude des ensembles flous, le raisonnement approximatif et à la commande par la logique floue. Nous proposons une commande floue pour contrôler le mouvement du robot manipulateur suivant une trajectoire joignant sa configuration initiale vers sa configuration finale.

La conception d'un planificateur de mouvement du bras manipulateur ainsi la discussion des résultats de simulation obtenus sont présentées au cinquième chapitre.

Le *chapitre six* est réservé à la conception d'un contrôleur de mouvement et la discussion des résultats de simulation de l'application du contrôleur proposé à la commande d'un robot manipulateur à deux degrés de liberté . La technique proposée est basée sur la commande séparée des mouvements du manipulateur et de la plate-forme mobile. La gestion commune du bras manipulateur et de la plate-forme mobile peut être envisagée.

1.3 Conclusion :

Les travaux dans le domaine de la manipulation mobile sont à ce jour nombreux et variés. D'après les nombreuses applications dans les différents domaines cités dans ce chapitre, il s'avère que de tels systèmes sont incontournables pour la réalisation de très nombreuses tâches. S'il est vrai que de tels systèmes offrent des possibilités intéressantes de locomotion et de manipulation, ils posent encore des problèmes au niveau de leur commande en raison de leur redondance et des différentes tâches qui peuvent leur être assignées. En présentant la synthèse de travaux réalisés, nous avons voulu donner un aperçu sur les travaux proposés par de nombreux chercheurs, et qui ne sont pas les moindres, qui ont contribué à des solutions liées à ce problème. Cependant, malgré le nombre important des travaux apparus dans ce domaine, il reste que de nombreux problèmes nécessitent plus d'attention quant à leur résolution.

Chapitre 2

La manipulation mobile

2.1. Introduction :

La robotisation est née à partir d'un besoin industriel où le nombre de tâches élémentaires et répétitives est élevé. Le robot a tendance à remplacer l'homme dans ses tâches répétitives et bien définies. Dans la partie commande, on trouve un simple asservissement articulaire du type PID, auquel on associe un générateur de trajectoire. Toutefois et malgré ses performances limitées, ce type d'architecture reste très répandu dans le milieu industriel.

Depuis, les robots manipulateurs ont gagnés en versatilité et en rapidité. Cela a été possible grâce au progrès technologique considérable et à la recherche scientifique qui ont pu apporter à cette discipline les moyens adéquats de nouveaux matériaux présentant légèreté et souplesse, une diversité de capteurs précis et d'actionnaires à rendements élevés. De l'autre côté, plusieurs commandes robustes développées dans le but de régler bon nombre de problèmes ont été implantés avec beaucoup de succès. Désormais, on assiste aujourd'hui à une robotisation à grande échelle intégrant des modules plus complexes de planification de trajectoires et d'évitement d'obstacles.

Cependant, ce succès considérable dans cette matière n'a pas laissé indifférent les chercheurs de penser à élargir l'espace de travail des bras manipulateurs. C'est pourquoi à l'image de l'homme qui possède la faculté de se déplacer pour réaliser des tâches à différents endroits ou pour effectuer des tâches continues nécessitent le déplacement du bras, des manipulateurs mobiles ont été envisagés. On peut ainsi définir un manipulateur mobile comme un bras manipulateur monté sur une plate-forme mobile. Cet assemblage, qui trouve de nombreuses applications dans les domaines terrestres, spatiales ou sous-marins, permet d'élargir le champ d'action du manipulateur et donc de réaliser des tâches de manipulation ou de portage sur un espace de travail plus important. La manipulation

mobile permet aussi de positionner le bras manipulateur dans une configuration plus confortable. L'objectif est de mettre en place la gestion des mouvements entre le bras manipulateur et le véhicule avec une coordination suffisante pour permettre la complémentarité des actions. Si la trajectoire du bras manipulateur est connue à priori dans un référentiel universel, alors le mouvement de la plate-forme peut être déterminé en conséquence. Cependant, si le mouvement de l'organe terminal n'est pas connu à priori, alors la planification de la trajectoire doit se faire localement et en ligne que globalement et hors ligne. Afin de mener le manipulateur mobile à exécuter une tâche spécifique avec une bonne précision, il est indispensable de construire un algorithme de contrôle pour le bras manipulateur ainsi que pour la plate-forme. Les approches existantes sur la manipulation mobile incluent les interactions dynamiques entre le véhicule et le bras manipulateur, la coordination des mouvements, ainsi que la coopération et la stabilité du système.

2.2. Stabilité des Manipulateurs Mobiles :

Les machines utilisées dans l'exécution des tâches dans les industries de construction ou dans les mines, pour ne citer que ceux là, utilisent une plate-forme mobile sur laquelle est placé un bras manipulateur. Pour effectuer des tâches de déplacement des charges, on envisage d'étudier le problème de la stabilité du système afin de gérer l'équilibre dynamique du système et prévenir le risque de renversement. Le manipulateur mobile peut avoir une bonne stabilité si la plate-forme mobile est suffisamment large que le bras manipulateur. Dans ce cas, la mobilité et la consommation de l'énergie deviennent deux inconvénients majeurs. Cependant, lorsque les dimensions de la plate-forme sont petites, il devient nécessaire d'étudier la stabilité du manipulateur mobile. On relève dans les travaux entrepris dans ce domaine que certains chercheurs se sont penchés sur ce problème pour étudier la stabilité statique et d'autres la stabilité dynamique. Cette dernière peut être vue comme une stabilité du système lorsque le véhicule est en mouvement ou la stabilité du système lors d'un mouvement simultané du bras manipulateur et de la plate-forme mobile. Mc Ghee et Frank [15] ont défini la marge de la stabilité statique comme la plus courte distance horizontale entre le centre de gravité et les limites du support du modèle. Ce support est défini comme la surface minimale du polygone convexe contenant tous les points de contact. Huang et al. [16], utilisent une méthode qui repose sur les différentes positions du bras par rapport à la base afin de modifier la position du centre de gravité de l'ensemble. La méthode utilisée est celle du ZMP (Zero Moment Point) qui

traduit l'équilibre dynamique du système et tient compte de la dynamique du manipulateur mobile ainsi que des perturbations dues à l'environnement. Messuri et Klein[17], introduisent une mesure différente de la stabilité appelée : Marge de stabilité d'énergie. Cette méthode a été étendue par Ghasempoor et Sepheri [18], pour inclure l'effet de tous les facteurs relatifs à la stabilité de la plate-forme. Qiang et al [19], abordent la stabilité du manipulateur mobile lorsque la plate-forme mobile et le bras manipulateur exécutent des mouvements simultanés. Ils dérivent le mouvement coordonné entre la plate-forme et le bras dans le suivi de trajectoires.

2.3. Les Applications des Manipulateurs Mobiles

2.3.1 Le domaine terrestre

Leur utilisation suscite un intérêt considérable aussi bien dans le milieu industriel que dans les milieux hostiles; tels que les centrales nucléaires. Dans les deux cas, le manipulateur mobile travaille conjointement avec l'homme ou le remplacer dans le cas défavorable comme par exemple le déminage.

2.3.2 Le domaine spatial

Leur utilisation dans ce domaine est surtout envisagée sur les sols planétaires où des relevés d'échantillons sont réalisés. Les manipulateurs mobiles peuvent aussi bien être utilisés dans la maintenance des stations orbitales.

2.3.3 Le domaine sous-marin

On peut voir aussi les manipulateurs mobiles utilisés dans le milieu sous-marin. Ils peuvent en effet contribuer à l'installation des câbles, le soudage des tuyaux ainsi qu'à l'exploration et la surveillance des sites.

2.4. Les Contraintes des Manipulateurs Mobiles

Le couplage entre la base mobile et le bras manipulateur offre plusieurs avantages. Néanmoins, certains problèmes apparaissent tels que :

- La contrainte de non-holonomie, qui interdit à la base mobile un déplacement latéral.
- L'interaction dynamique entre la plate-forme et le bras manipulateur crée des couples perturbateurs qui peuvent donner de faibles performances au bras manipulateur.
- L'interaction roue-sol est susceptible de produire des dérapages ou des glissements.

- La redondance apportée par la mobilité de la base et qui en résulte en plusieurs configurations du manipulateur mobile pour un même point.

Toutes ces contraintes doivent être prises en considération dans la commande si de bonnes performances sont souhaitables.

2.5. Commande des Manipulateurs Mobiles

La commande des manipulateurs mobiles fait émerger la notion de coordination de mouvements entre la locomotion et la manipulation, la coopération du bras et de la base a suscité de nombreuses publications, et s'est rapidement posé le problème de définir un critère permettant d'évaluer quantitativement la coordination entre locomotion et manipulation. Ainsi les concepts de manipulabilité du bras (Yoshikawa,) [20] et de mobilité de la plate-forme ont été regroupés dans la notion de dextérité du manipulateur mobile (Foulon et al) [21], (Bayle et al) [22], (Tchon et Muszynski) [23], (Tchon et Zadarnowska,) [24], qui fournit un critère dont l'optimisation permet un contrôle coordonné de la plate-forme et du bras (Bayle) [25]. Les travaux sur les manipulateurs mobiles holonomes (Papadopoulos et Dubowsky)[26], (Khatib et al) [14], assimilables à des bras manipulateurs de dimension étendue, ont fait apparaître une hiérarchie naturelle entre la plate-forme omnidirectionnelle aux dynamiques lentes, et le bras embarqué au positionnement plus précis et aux dynamiques plus rapides. La stratégie de coordination repose sur une relation de type maître / esclave (Luksch et al) [27] entre manipulation et locomotion. Il s'agit d'utiliser la locomotion afin de maintenir les variables articulaires du bras à mi-chemin de leurs butées [14]. Cette notion de situation préférentielle de la plate-forme mobile a également été employée dans le cadre de manipulateurs mobiles non-holonomes [10] pour maintenir le manipulateur autour d'une configuration où sa manipulabilité est importante.

Une part importante de la littérature est dédiée aux méthodes de planification de trajectoire, c'est-à-dire aux approches de prédétermination d'une solution prenant en compte la structure de l'espace de travail de l'organe terminal et les contraintes sur la plate-forme. Les méthodes employées pour résoudre le problème sont variées : méthode d'optimisation sous contrainte (Perrier et al) [28],(Furuno et al) [29], notamment du critère de dextérité dans (Bayle et al) [30] ou dans le cadre de la commande optimale (Desai et al) [31] . Notons l'apport du concept de bande élastique proposé par (Quinlan et al) [32] pour les plates-formes omnidirectionnelles, étendu au manipulateurs mobiles (Brock et Khatib,)[33] et aux plates-formes non-holonomes (Laminaux et Bonnafous) [34], qui propose un lien entre

planification et commande. Cette approche permet, en fonction de la présence éventuelle d'obstacles, de déformer en temps réel une trajectoire planifiée par avance, ce qui pallie le manque de réactivité des méthodes de planification classiques.

Etant donné qu'il est généralement impossible de stabiliser asymptotiquement la **situation complète d'une plate-forme non-holonome le long d'une trajectoire de référence** quelconque, les lois de commande par retour d'état proposées jusqu'ici dans la littérature conduisent à un contrôle partiel de l'état du manipulateur mobile (plus précisément un contrôle de la situation de l'effecteur du manipulateur et d'une partie de la situation de la plate-forme). Les contributions en ce sens sont par exemple les travaux de (Yamamoto et Yun,)[10] pour une commande en position de la plate-forme par linéarisation entrée-sortie, (Egerstedt et Hu,) [35] pour une commande plus nettement découplée où le contrôle de la plate-forme repose sur le suivi d'un chemin de référence (Canudas de Wit et al.) [36], ou encore (Miksch et Schroeder) [37] et (Wang et Kumar, 1993)[38] pour une approche de minimisation fonctionnelle sous contrainte via les conditions d'optimalité de Lagrange. Citons aussi l'utilisation de l'optimisation sous contrainte du critère de dextérité utilisé dans (Bayle et al.) [22]. Des approches héritées du formalisme des bras manipulateurs, sans découplage entre base mobile et bras manipulateur, ont également été développées dans (Bayle et al.) [39] et (Seraji,) [40] avec l'utilisation de tâches additionnelles. Toutes ces méthodes s'appuient sur le fait qu'un contrôle partiel de la situation de la base peut se montrer suffisant pour les tâches de manipulation considérées. Cependant, dès que la stratégie de coordination impose à la plate-forme d'effectuer des manœuvres (par exemple afin d'éviter les butées articulaires du manipulateur), un contrôle de la situation complète de la plate-forme est nécessaire. C'est la raison pour laquelle une approche pour la commande des manipulateurs mobiles requiert de commander la plate-forme à la fois en position et en orientation, problème qui, à notre connaissance, n'a pas été traité jusqu'à présent.

On précise aussi que de nombreux travaux ont déjà traité la modélisation géométrique et cinématique des manipulateurs mobiles, c'est-à-dire de systèmes composés d'une plate-forme mobile porteuse d'un bras manipulateur. Nous renvoyons donc le lecteur aux études ('Canudas de Wit et al) [36] , (Campion et al.) [41] et (Neimark et Fufaev,)[42] pour un état de l'art sur la modélisation des plates-formes mobiles à roues et de leurs cinématiques, et (Bayle et al.) [22] pour la modélisation des manipulateurs mobiles et les modèles géométriques directs des manipulateurs.

Dans le cadre de cette étude, une modélisation des manipulateurs mobiles très générale et détaillée n'est pas nécessaire puisque l'on se concentrera principalement sur le contrôle du bras manipulateur en mode autonome.

L'analyse du système mécanique abordé dans notre thèse est donc constitué d'une base mobile non-holonome sur laquelle est montée un robot manipulateur à 2 d.d.l, formant ainsi le manipulateur mobile. Il a été montré aux paragraphes précédents que seul le contrôle du bras manipulateur en mode autonome a été abordé, dans cette optique, nous allons décrire la constitution et modélisation des robots manipulateurs.

2.6. Constitution et modélisation des robots manipulateurs.

2.6.1. Définition d'un robot manipulateur :

Parmi les différentes définitions d'un robot manipulateur qu'on trouve dans la littérature, nous avons retenu [43] :

Celle donnée par le dictionnaire : un robot est une machine à l'aspect humain, capable de se mouvoir, de manipuler des objets ou d'exécuter des opérations selon un programme fixe ou modifiable.

Celle donnée par la JIRA (Japan Industrial Robot Association) : c'est un système versatile doté d'une mémoire, capable d'effectuer des mouvements comme ceux d'un opérateur humain.

Celle donnée par RIA (Robot Institut of America) : c'est un manipulateur à fonction multiple pouvant être programmé pour réaliser automatiquement des tâches variées, éventuellement répétitives.

La définition descriptive d'un robot : un robot manipulateur est caractérisé par une structure arborescente articulée simple ou multiple dont les segments sont mobiles les uns par rapport aux autres. Cet ensemble a pour objectif de ramener l'organe terminal vers un lieu géométrique imposé par la tâche.

2.6.2 Constitution d'un robot manipulateur :

Un robot manipulateur est l'ensemble formé par (Figure 2.1) [43] :

- Une structure mécanique qui supporte l'organe terminal à situer.
- Des actionneurs qui servent à agir sur la structure précédente pour en modifier la configuration et donc la situation de l'organe terminal.
- Des capteurs divers nécessaires à la commande, parmi lesquels on distingue les capteurs proprioceptifs qui permettent la saisie de l'état mécanique du robot manipulateur.

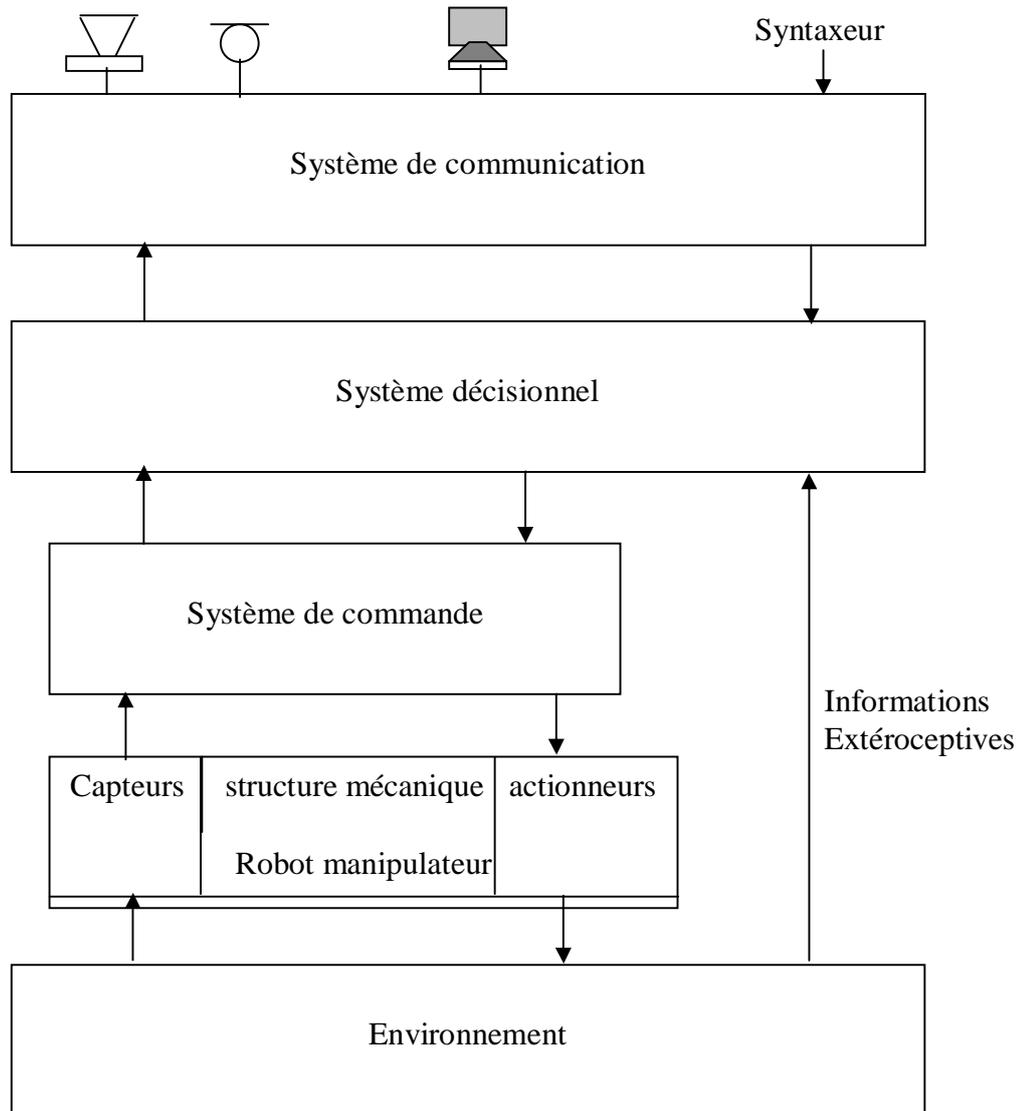


Figure 2-1: Structure générale d'un robot manipulateur.

Les capteurs extéroceptifs qui indiquent l'état de l'environnement du robot manipulateur.

- Un système de commande qui pilote les actionneurs du robot manipulateur à partir de la définition du mouvement à exécuter transmise par le système décisionnel et des informations délivrées par les capteurs proprioceptifs.
- Un système décisionnel qui assure la fonction de raisonnement et élabore le mouvement du robot manipulateur à partir de la définition de la tâche à exécuter transmise par l'opérateur à l'aide du système de communication. Les fonctions de base du système décisionnel sont

l'interprétation et la compréhension des messages émis par l'opérateur, la résolution des problèmes posés, la génération de plans et la génération de messages à destination de l'opérateur ainsi que la gestion de la base de données nécessaire à ce système.

- Un système de communication qui gère les messages transmis entre le système décisionnel et l'opérateur via une console de visualisation alphanumérique ou graphique, un système d'analyse et de synthèse vocale, un syntaxeur, ...etc.

2.6.3. Structure mécanique d'un robot manipulateur :

2.6.3.1. Mécanisme :

La partie mécanique d'un robot industriel est constituée de deux (02) sous ensembles distincts [44] :

- Un (ou plusieurs) organe terminal : considéré comme une interface permettant au robot d'interagir avec son environnement (poignet, préhenseur, effecteur, ...).
- Une structure mécanique articulée : qui sert à ramener l'organe terminal dans une position et orientation données, selon certaines caractéristiques de vitesse et d'accélération.

La structure mécanique d'un robot est une chaîne cinématique de corps rigides (ou souples) assemblés par des liaisons appelées articulations, les chaînes peuvent être soit ouvertes simples, soit arborescentes ou fermées.

a- Structure à chaîne simple ouverte (structure sérielle) : Constituée par une succession d'articulations depuis la base jusqu'à l'organe terminal (Figure 2.2)

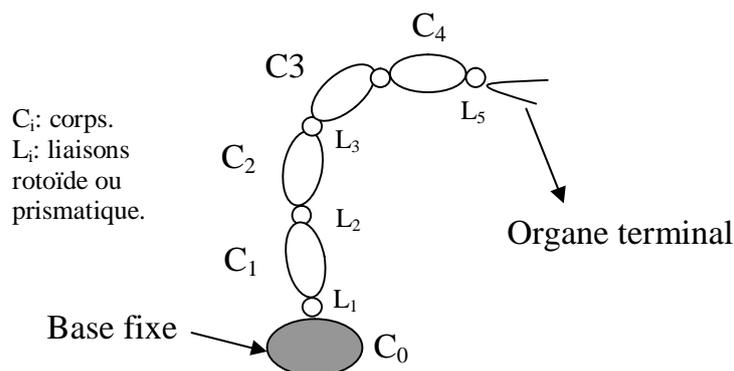


Figure 2.2 Structure à chaîne simple ouverte.

b- Structure à chaîne arborescente : Possède deux ou plusieurs chaînes simples partant d'une même chaîne commune, chacune pouvant avoir un organe terminal (Figure 2.3).

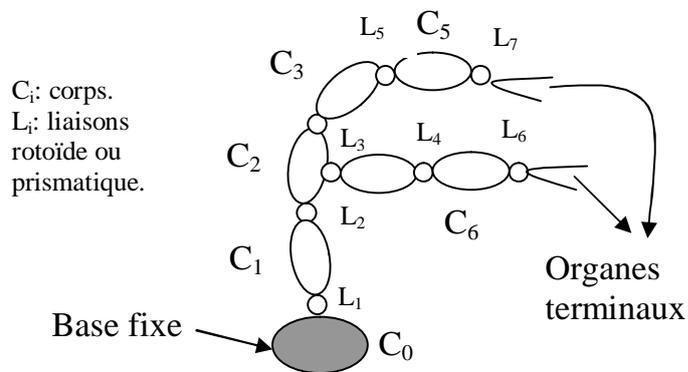


Figure 2.3 Structure à chaîne arborescente.

c- Structure à chaîne complexe (structure parallèle) : constituée de boucles mécaniques fermées, c'est à dire, la majorité des articulations se trouvent en parallèle les unes par rapport aux autres (Figure 2.4).

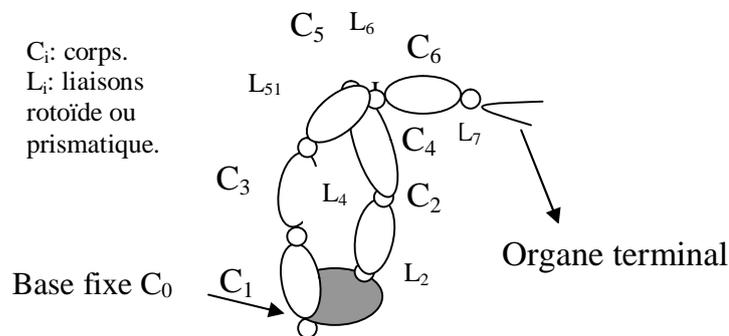


Figure 2.4: Structure à chaîne complexe (fermée).

La plupart des robots utilisés actuellement dans l'industrie sont à chaîne simple ou parfois arborescente, mais les structures parallèles sont en phase de recherche.

2.6.3.2. Articulation :

C'est une liaison mécanique entre deux corps successifs limitant le nombre de degrés de liberté (d.d.l) de l'un par rapport à l'autre.

$0 < m \leq 6$ où m est le degré de liberté (ou de mobilité).

si $m = 1$ (le cas le plus fréquent en robotique) l'articulation est :

- Soit rotoïde (notée R).
- Soit prismatique (notée P).

a. Articulation rotoïde:

Il s'agit d'une articulation de type pivot réduisant le mouvement entre deux corps à une rotation autour d'un axe commun, la situation relative entre les deux corps est donnée par l'angle (notée θ) autour de cet axe (Figure 2.5.a).

b. Articulation prismatique :

C'est une articulation de type glissière réduisant le mouvement entre deux corps à une translation le long d'un axe commun, la situation entre les deux corps est mesurée par la distance (notée L) le long de cet axe (Figure 2.5.b).

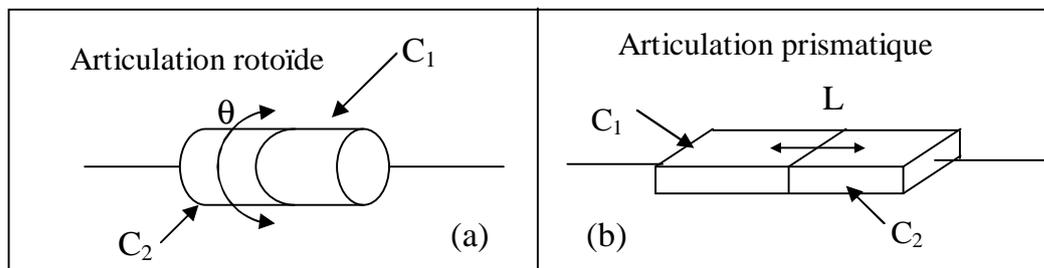


Figure 2.5 : Symboles des liaisons rotoïde et prismatique

On peut constituer des liaisons de mobilité supérieure à 1 en combinant des articulations simples. Une articulation cylindrique (c), par exemple, est obtenue par (R+P), une articulation sphérique (s) est obtenue avec $3 \times R$ à axes concourants.

2.6.3.3. Degré de liberté d'un robot manipulateur :

Le degré de liberté (d.d.l) d'un robot manipulateur est égal au nombre de paramètres indépendants qui fixent la situation de l'organe terminal où l'inégalité suivante est toujours vérifiée : $d.d.l \leq m$.

Où m est l'indice de mobilité de la structure mécanique.

2.6.3.4. Degré de liberté d'une tâche :

Le degré de liberté d'une tâche (d.l.t) est égal au nombre de paramètres indépendants qui permettent de fixer toutes les situations que l'organe terminal doit atteindre. Pour que le robot manipulateur puisse exécuter une tâche donnée, la condition $d.l.t \leq d.d.l$ est nécessaire mais pas suffisante [45].

2.6.3.5. Redondance d'un robot manipulateur :

Un robot est redondant lorsque le nombre de d.d.l de l'organe terminal est inférieur au nombre d'articulations motorisées. Cette propriété permet d'augmenter le volume du domaine accessible et de préserver les capacités de déplacement de l'organe terminal en présence d'obstacles (le ou les d.d.l. supplémentaires permettant leur contournement).

Pour les chaînes simples ouvertes, on peut citer les combinaisons suivantes qui donnent une structure redondante :

- Plus de six articulations motorisées.
- Plus de trois articulations rotoïdes d'axes concourants.
- Plus de trois articulations rotoïdes d'axes parallèles.
- Plus de trois articulations prismatiques.
- Deux axes d'articulations prismatiques parallèles.
- Deux axes d'articulations rotoïdes confondus.

2.6.3.6. Espace articulaire (espace de configuration) :

C'est l'espace des variables (ou coordonnées) articulaires, noté \mathfrak{R}^n , qui permet de décrire l'état de la situation des différents corps du robot, la dimension "n" est égale au nombre de variables articulaires indépendantes et correspond au nombre de degrés de liberté de la structure mécanique.

2.6.3.7. Espace opérationnel :

Cet espace, noté \mathcal{R}^m , est celui dans lequel est représentée la situation de l'organe terminal. La dimension "m", qui représente le nombre de d.d.l. maximal que peut avoir cet organe, est égale au nombre de paramètres indépendants nécessaires pour décrire la situation de l'organe terminal dans l'espace cartésien (coordonnées cartésiennes).

$$m \leq 6 \text{ (rotation ou translation) et } m \leq n.$$

2.6.3.8. Volume de travail d'un robot manipulateur :

Le volume de travail d'un robot manipulateur est défini comme l'espace engendré par un point de l'organe terminal lorsque la configuration du robot évolue. Le choix du point précédemment évoqué est arbitraire, certains industriels considèrent le centre du poignet s'il existe et obtiennent donc un volume de travail minimal, d'autres prennent le point situé à l'extrémité de l'organe terminal ou encore le centre de la platine de fixation de celui-ci qui existe sur tous les modèles. Ceci démontre la relative imprécision qui affecte la définition de cette caractéristique, de plus, les orientations de l'organe terminal qu'il est possible d'obtenir en un point du volume de travail, fonction de configuration, n'apparaissent pas dans la définition de ce volume de travail. Ce dernier, bien qu'étant une caractéristique approximative de la performance d'un robot manipulateur permet cependant de comparer les différentes structures existantes [43].

2.6.4. Les actionneurs :

Les organes d'un système robotisé sont mis en mouvement par des dispositifs d'entraînement qui convertissent en déplacements les signaux de commande qui leur ont été fournis.

Les dispositifs d'entraînement asservis couramment utilisés par les robots sont [45].

2.6.4.1. Les actionneurs hydrauliques :

Les actionneurs hydrauliques sont utilisés sur les robots de forte puissance du fait qu'ils fournissent une puissance nominale importante sous un volume réduit.

La figure 2.6 schématise l'organisation d'une installation hydraulique.

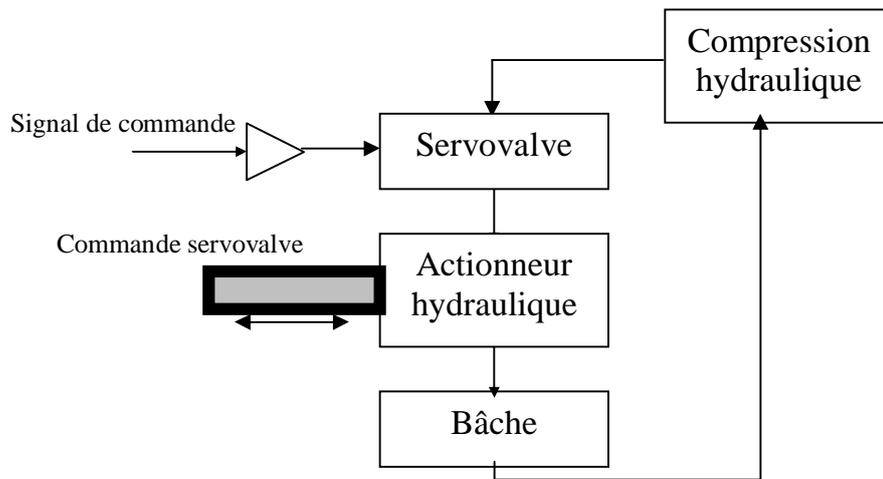


Figure 2.6: Organisation d'une installation hydraulique

2.6.4.2. Les actionneurs électriques :

La plupart des robots équipés d'actionneurs électriques (>85%) utilisent des moteurs à courant continu [46].

a- Les moteurs à courant continu :

Ils sont caractérisés par le fait que le champ inducteur occupe une direction fixe, ce champ peut être créé de deux façons :

- Par un aimant permanent,
- Par un circuit électromagnétique.

Deux modes de commande sont alors possibles :

- Les moteurs à commande d'inducteur : l'inducteur est alimenté par une tension variable qui constitue la commande du moteur, tandis que l'induit est parcouru par une intensité constante. Les caractéristiques couple-vitesse de rotation de ce mode de commande se présentent sous la forme d'une famille de droites horizontales comme l'indique la figure 2.7.

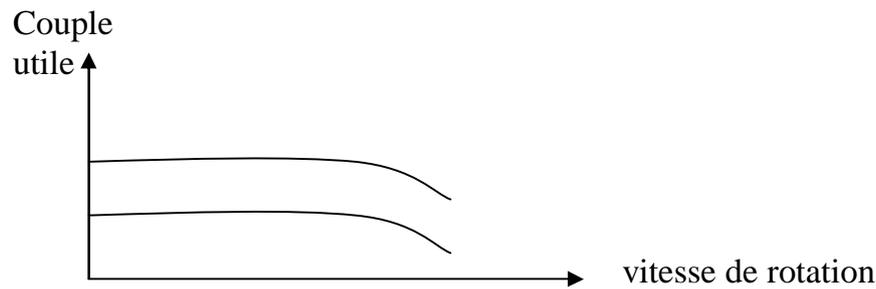


Figure 2.7 : Evolution du couple en fonction de la vitesse. Commande d'inducteur

- Les moteurs à commande d'induit : dans ce type, l'inducteur est alimenté à intensité constante, tandis que l'induit est alimenté par une tension variable qui constitue la commande du moteur. L'allure des courbes caractéristiques de ce mode de commande est assez différente de celui à commande d'inducteur comme le montre la figure 2.8.

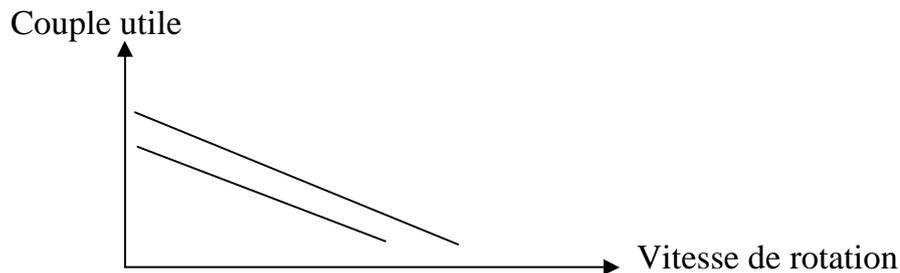


Figure 2.8 : Evolution du couple en fonction de la vitesse. Commande d'induit

b-Les moteurs pas à pas :

Dans ce type de moteur, la rotation de l'arbre s'effectue par pas de façon discrète contrairement aux moteurs décrits précédemment. Chaque pas a la même amplitude.

Ce type de moteur est composé de deux parties :

- Le stator qui est pourvu de bobinages répartis également sur la circonférence,
- Le rotor peut être, en général, réalisé selon deux technologies différentes : à aimant permanent ou à empilage de tôles encochées.

Le principe de fonctionnement de ce type de moteur est schématisé sur la figure 2.9.

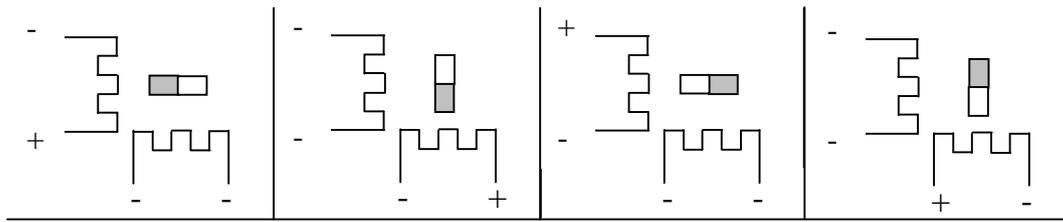


Figure 2.9: Principe de fonctionnement d'un moteur pas à pas.

2.6.5. Les capteurs :

La découverte de base de l'automatique qui date depuis la seconde guerre mondiale réside dans le concept de boucle de régulation qui signifie que la sortie d'un système réagit sur son entrée par la différence entre la sortie désirée et la sortie réelle. Pour qu'on puisse mettre en œuvre de tels systèmes, il faut évidemment pouvoir mesurer la valeur de cette différence, c'est le rôle du capteur [43][45].

Tous les robots comportent deux grandes boucles de régulation, la première correspond à celle que l'on rencontre sur les machines automatiques classiques, cette boucle considère le robot indépendamment de son environnement et de la tâche véritable qu'il réalise. Les capteurs impliqués vont être des capteurs internes ou proprioceptifs permettant un contrôle permanent de la bonne exécution du mouvement.

La seconde boucle de régulation est liée à l'exécution de la tâche et intervient en amont de la boucle interne, elle va impliquer des capteurs permettant de voir, de toucher, de comprendre l'environnement et la tâche de telle manière que le robot puisse définir tout seul quelles trajectoires emprunte pour exécuter sa tâche. Une fois les trajectoires sont déterminées, la première boucle de régulation prend le relais pour s'assurer du bon suivi de celles-ci. Les capteurs impliqués dans la seconde boucle seront les capteurs externes ou extéroceptifs du robot, ceux qui le renseignent sur l'état de situations liées à l'environnement.

Il existe aussi une troisième classe de capteurs dits capteurs de sécurité.

2.6.5.1. Les capteurs internes (proprioceptifs) :

a- Les capteurs de position ou de déplacement :

On sait que sur un robot, les corps se déplacent les uns par rapport aux autres soit par des rotations soit par des translations, le rôle de ce genre de capteurs est de mesurer ces déplacements. La figure 2.10 résume les divers types de capteurs rencontrés en robotique ou on distingue deux types de capteurs [45] :

- Capteurs de mouvements rectilignes (potentiomètres, transformateur variable, ...).
- Capteurs de déplacements angulaires (optiques, magnétiques, ...).

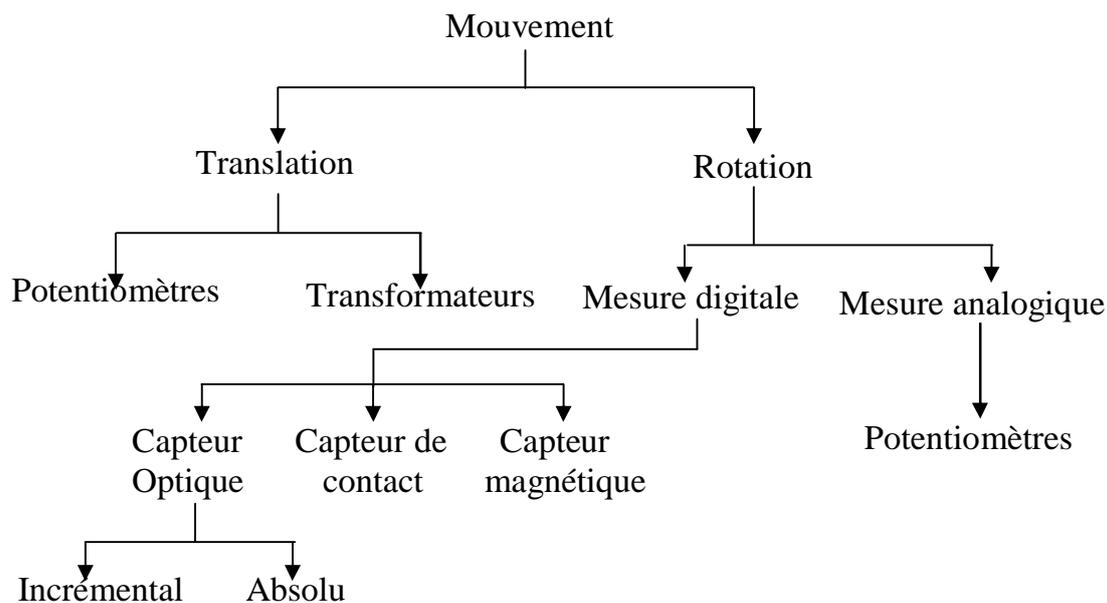


Figure 2.10: Les principaux capteurs de position rencontrés en robotique.

b- Les capteurs de vitesse :

Comme il est nécessaire de savoir mesurer les déplacements en translation et en rotation, il est aussi important de savoir à quelle vitesse ces déplacements ont eu lieu.

Dans la majorité des cas on essaie de se ramener à la mesure d'une vitesse de rotation car la mesure d'une vitesse de translation nécessite des capteurs très spéciaux, peu nombreux et de toute façon, rarement utilisés.

c- Les capteurs d'effort :

Ils sont moins utilisés que les capteurs de position ou de vitesse en tant que capteurs internes, ils sont par contre très utilisés comme capteurs externes pour le contrôle d'effort, par exemple : capteur d'effort piézo-électrique, capteur d'accélération utilisé surtout dans la commande dynamique.

2.6.5.2. Les capteurs externes (extéroceptifs) :

Les capteurs externes ne sont liés qu'à la tâche utile que doit réaliser le robot, cette tâche peut varier à l'infini et on sait très mal décrire une tâche inconnue à priori en termes de grandeurs à mesurer. En conséquence la question qu'on se pose est la suivante : pour exécuter une tâche quelconque, quelles fonctions informatisées vont être impliquées ? . Se référant à l'homme et à ses "sens", on répond immédiatement que la vision et le sens tactile sont les fonctions les plus riches pour exécuter correctement une tâche donnée. Un capteur de fonction dépasse très loin la notion usuelle de capteur, ce doit être une machine tout à fait complexe avec une partie mesure et une partie logicielle (Figure 2.11).

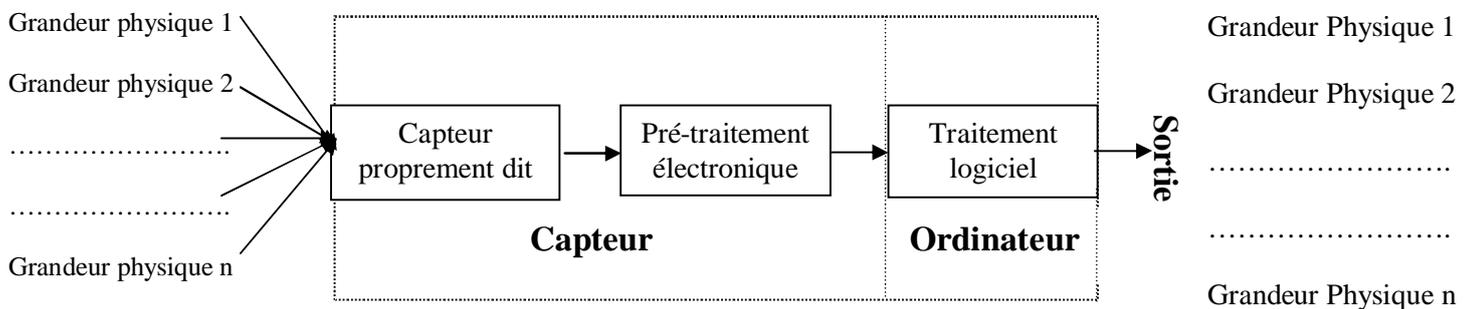


Figure 2.11: Les différentes parties d'un capteur externe.

Usuellement les deux fonctions que l'on a cherché à établir pour les robots sont la fonction tactile (le toucher) et la fonction visuelle (la vision).

a. Les capteurs de la fonction tactile :

Ce genre de capteurs est utilisé dans la robotique pour obtenir des informations sur le contact entre la main du robot et les objets manipulés, ces informations peuvent être utilisées, par exemple, pour la localisation et la reconnaissance des objets ainsi pour le contrôle de la force exercée par le robot sur l'objet à manipuler.

b. Les capteurs de la fonction visuelle :

Contrairement aux capteurs tactiles qui exigent un contact avec l'objet sur lequel ils vont faire des mesures, les capteurs visuels prélèvent une information à distance. Le support de cette information est un rayonnement qui peut être thermique, sonore, électromagnétique ...etc. Pour les robots industriels, les capteurs de la fonction visuelle souvent utilisés sont : les capteurs ultrasonores, les capteurs à infrarouge et même les caméras.

2.7. Modélisation des robots manipulateurs :

2.7.1. Introduction :

Pour effectuer l'analyse et la synthèse d'un système dynamique, il est nécessaire de connaître les relations entre ses grandeurs d'entrées et de sorties, l'ensemble de ces relations constitue le modèle mathématique du système. L'efficacité de ce modèle repose sur une analogie entre le comportement des objets physiques et celui des êtres mathématiques [44].

Dans le cadre de l'automatique, modéliser un système consiste à établir un ensemble de relations mathématiques qui permettent de décrire, avec une précision suffisante, les interactions entre ce système et son environnement extérieur. Lorsque les relations suscitées sont issues des équations de la physique, le modèle obtenu est dit modèle de connaissance, si ces relations découlent des observations disponibles sur le système, on aboutit ainsi au modèle de présentation obtenu par identification.

Dans le domaine de la robotique l'élaboration du modèle nécessite une étude approfondie et détaillée de la structure du robot.

2.7.2. Modélisation géométrique :

Un robot manipulateur quelle que soit la fonction qui lui est attribuée (manipulation, assemblage, soudage, ...), doit situer dans l'espace un objet appelé organe terminal. Il peut s'agir d'une pince maintenant une pièce, d'un pistolet de peinture ou d'un autre type d'outil. La situation de cet organe terminal, en général quelconque et évolutive, est définie par la position de l'un de ses points et par son orientation autour de ce point.

2.7.2.1. Représentation d'un point dans l'espace :

Chaque point dans l'espace peut être représenté par un vecteur colonne à quatre composantes. Les trois premières sont les coordonnées cartésiennes du point, tandis que la quatrième composante est considérée comme un facteur d'échelle, égal à 1 en robotique.

$$V = [V_x, V_y, V_z, 1]^T \quad (2.1).$$

2.7.2.2. Représentation d'un vecteur dans l'espace :

Pour représenter un vecteur dans l'espace, on se sert de quatre composantes. Trois pour les coordonnées cartésiennes et la quatrième égale à zéro (0).

$$U = [U_x, U_y, U_z, 0]^T \quad (2.2).$$

2.7.2.3. Représentation d'un plan dans l'espace :

Le plan $\alpha.x + \beta.y + \gamma.z + \delta = 0$, est représenté par un vecteur Q donné par :

$$Q = [\alpha, \beta, \gamma, \delta]. \quad (2.3).$$

2.7.2.4. Transformation homogène :

a. Transformation des repères :

Toute transformation, translation et /ou rotation, d'un repère R_i qui l'amène sur le repère R_j est définie par la matrice ${}^i T_j$ de dimension (4x4).

$${}^i T_j = \begin{bmatrix} s_x & n_x & a_x & p_x \\ s_y & n_y & a_y & p_y \\ s_z & n_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = [s \quad n \quad a \quad p] \quad (2.4).$$

Avec:

- s, n, a : sont les vecteurs unitaires du repère R_j exprimé dans le repère R_i .
- p : est l'origine du repère R_j exprimé dans le repère R_i .

Matrice de transformation de translation pure :

La translation d'un repère R_i vers un repère R_j est représenté par une matrice ${}^i T_j$ notée $\text{trans}(x, d_1)$, $\text{trans}(y, d_2)$ ou $\text{trans}(z, d_3)$, avec d_i est la longueur de translation.

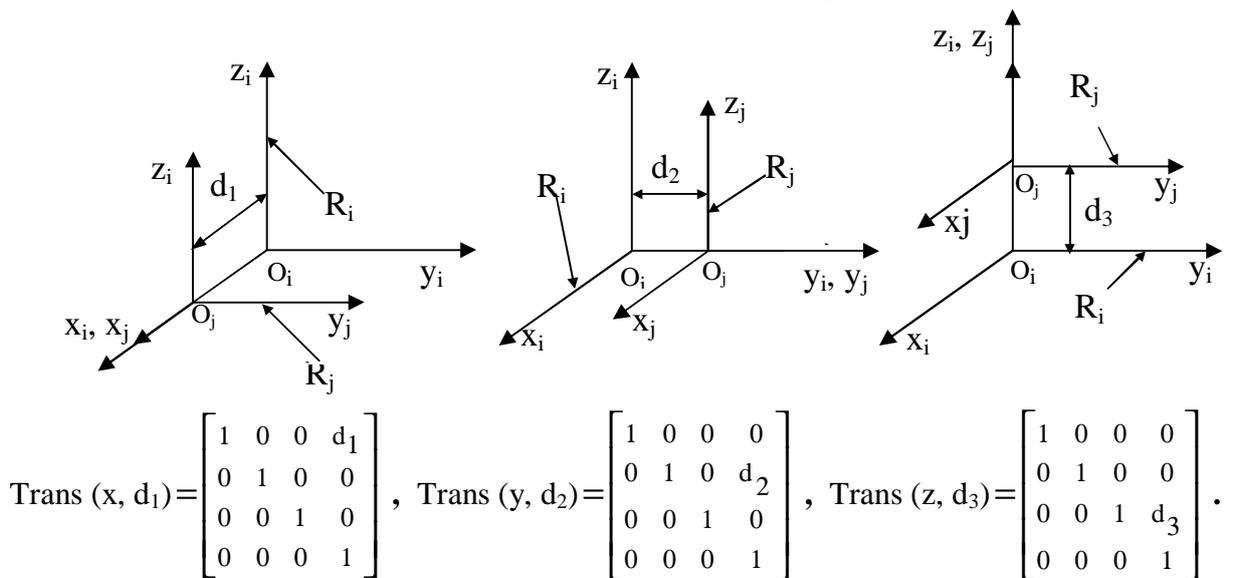


Figure 2.12: Transformation de translation pure.

- Matrice de transformation de rotation pure :

La rotation d'un repère R_i vers un repère R_j est représentée par une matrice ${}^i T_j$ notée $\text{rot}(x, \theta_1)$, $\text{rot}(y, \theta_2)$ ou $\text{rot}(z, \theta_3)$, avec θ_i est l'angle de rotation.

Pour une rotation de θ_1 autour de l'axe 'x', la matrice de transformation est donnée par :

$$\text{rot}(x, \theta_1) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\theta_1 & s\theta_1 & 0 \\ 0 & -s\theta_1 & c\theta_1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

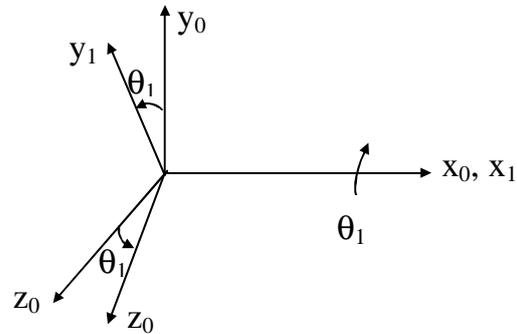


Figure 2.13: Transformation de Rotation pure.

Remarque : dans le cas de plusieurs transformations, quel que soit leur type, la matrice de transformation finale est le produit de ces matrices de transformation faites.

2.7.2.5. Modèle géométrique direct :

C'est l'ensemble des équations mathématiques qui permettent le passage de l'espace articulaire vers l'espace opérationnel, c'est à dire, les équations qui permettent d'obtenir les coordonnées cartésiennes de l'organe terminal sachant les variables articulaires.

Pour cela, il existe plusieurs techniques, parmi elles celle de 'DENAVID HARTENBERG' qui est donnée comme suit [44] :

Pour passer de O_{n-1} à O_n et de R_{n-1} à R_n (Figure 2.14), on peut toujours suivre le chemin suivant :

- 1) Une rotation autour de z_{n-1} d'un angle θ : $\text{rot}(z_{n-1}, \theta)$.
- 2) Une translation le long de z_{n-1} d'une longueur d : $\text{trans}(z_{n-1}, d)$.
- 3) Une translation le long de x_n d'une longueur a : $\text{trans}(x_n, a)$.
- 4) Une rotation autour de x_n d'un angle α : $\text{rot}(x_n, \alpha)$.

Les grandeurs θ , α , a et d sont définies de la façon suivante :

d : Distance de O_{n-1} à la projection de O_n sur l'axe z_{n-1} ,

a : Distance de O_{n-1} à la projection de O_{n-1} sur le plan $y_n O_n z_n$,

θ : Angle compris entre x_{n-1} et le plan défini par l'axe z_{n-1} et le point O_n ,

α : Angle formé par la projection de z_{n-1} sur le plan normal au plan (z_{n-1}, O_n) et l'axe z_n .

La forme générale de la matrice faisant passer d'une articulation à la suivante sera donc toujours :

$${}^nT_{n-1} = \text{Rot}(z_{n-1}, \theta) \times \text{trans}(z_{n-1}, d) \times \text{trans}(x_n, a) \times \text{rot}(x_n, \alpha)$$

$$= \begin{bmatrix} c\theta & -s\theta.c\alpha & s\theta.s\alpha & a.c\theta \\ s\theta & c\theta.c\alpha & -c\theta.s\alpha & a.s\theta \\ 0 & s\alpha & c\alpha & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.5).$$

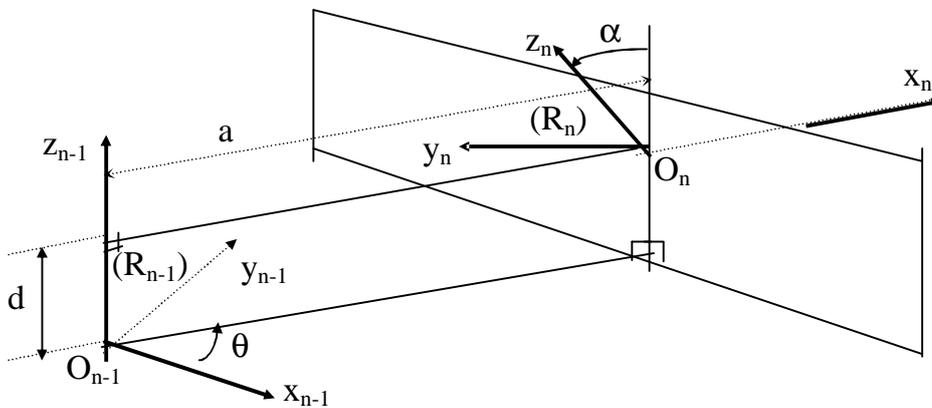


Figure 2.14: Représentation graphique de la méthode de D.H

La matrice ${}^nT_{n-1}$ représente la matrice de passage du repère R_{n-1} au repère R_n , pour calculer le modèle géométrique final on doit multiplier toutes les matrices de transformation élémentaires faites.

2.7.2.6. Modèle géométrique inverse :

Ce modèle est appelé modèle géométrique pour la commande, il permet de calculer les coordonnées articulaires en fonction des coordonnées opérationnelles et s'écrit sous la forme :

$$q = f^{-1}(x) = g(x) \quad (2.6).$$

Soit ${}^0T_n = {}^0T_1 \times {}^1T_2 \times \dots \times {}^{n-1}T_n$, la matrice de transformation homogène finale.

On note :

$$U_0 = \begin{bmatrix} s_x & n_x & a_x & p_x \\ s_y & n_y & a_y & p_y \\ s_z & n_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.7).$$

U_0 correspond à la situation désirée (donnée) de l'organe terminal, et par comparaison des éléments des matrices, on peut calculer les variables articulaires q_i en fonction des coordonnées cartésiennes, ce qui donne le modèle géométrique inverse [44].

2.7.3. Modélisation cinématique :

2.7.3.1. Modèle cinématique (différentiel) Direct (MCD) :

Ce modèle permet de déterminer la vitesse de l'organe terminal dans l'espace opérationnel en fonction des vitesses articulaires.

Nous l'obtenons, en dérivant par rapport au temps les coordonnées opérationnelles et les coordonnées articulaires :

$$\dot{X} = J(q) \times \dot{q} \quad (2.8)$$

où $\dot{q} = (\dot{q}_1, \dot{q}_2, \dots, \dot{q}_n)$ (2.9)

Et $J(q)$: matrice jacobienne de dimension $(m \times n)$ et s'écrit sous la forme :

$$J(q) = [J_{ij}] \quad (2.10)$$

Avec $J_{ij} = \partial f_i(q) / \partial q_j$; où $i=1, 2, \dots, m$ et $j=1, 2, \dots, n$ (2.11)

2.7.3.2. Modèle cinématique inverse (MCI) :

Ce modèle nous permet de calculer les vitesses articulaires \dot{q} en fonction des vitesses cartésiennes \dot{X} :

$$\dot{q} = J^{-1}(q) \times \dot{X} \quad (2.12)$$

où :

$$J^{-1}(q) = \text{adj}(J(q)) / |\det(J)|; \text{adj}(J(q)) \text{ est la matrice adjointe du jacobien.}$$

2.7.3.3. Modèle cinématique du 2^{ème} ordre (direct et inverse) :

Ce modèle permet de déterminer les relations entre les accélérations articulaires et cartésiennes :

$$\text{- direct : } \ddot{X} = J(q) \times \ddot{q} + \dot{J}(q) \times \dot{q} \quad (2.13)$$

$$\text{- inverse : } \ddot{q} = J^{-1}(q) \times (\ddot{X} - \dot{J}(q) \times \dot{q}) \quad (2.14)$$

$$\text{Où : } \dot{J}(q) = [\dot{J}_{ij}]_{ij}, \text{ avec } \dot{J}_{ij} = dJ_{ij} / dt = \sum_{k=1}^n \partial J_{ij} / \partial q_k \quad (2.15)$$

2.7.4. Modélisation dynamique d'un robot manipulateur :

Le modèle géométrique nous a permis de mettre en évidence les relations entre les dimensions du robot et ses différentes caractéristiques, en plus des masses et inerties des corps du robot, les dimensions géométriques auront également une influence sur le comportement dynamique des éléments mobiles et sur les couples nécessaires aux moteurs [47].

Le modèle dynamique détermine la relation entre les couples (et/ou forces) appliqués aux actionneurs et les positions, vitesses et accélérations articulaires, et s'écrit sous la forme suivante :

$$\Gamma = f(q, \dot{q}, \ddot{q}, F) \quad (2.16)$$

Avec :

Γ : Vecteur des couples/ forces des actionneurs selon l'articulation (rotoïde ou prismatique).

q : Vecteurs des positions articulaires.

\dot{q} : Vecteur des vitesses articulaires.

\ddot{q} : Vecteur des accélérations articulaires.

F : Efforts extérieurs (forces ou couples) exercés sur le robot manipulateur.

Il convient donc d'appeler modèle dynamique inverse ou tout simplement modèle dynamique, la relation de la forme (2.16). Le modèle dynamique direct est celui qui permet

d'exprimer les accélérations en fonction des vitesses, des positions, des couples et des conditions initiales du mouvement.

$$\ddot{\mathbf{q}} = \mathbf{g}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{\Gamma}, \mathbf{F}) \quad (2.17)$$

Pour déterminer ce modèle, il existe plusieurs formalismes ; pour notre cas, on retient le formalisme de LAGRANGE qui décrit les équations de mouvement en terme de travail et énergie du système :

$$\Gamma_i = d(\partial L / \partial \dot{q}_i) / dt - \partial L / \partial q_i, \quad i = 1, 2, \dots, n \quad (2.18)$$

Où :

L : Le lagrangien = E - P.

E : Energie cinétique totale du robot manipulateur.

P : Energie potentielle totale du robot manipulateur.

2.7.4.1. Calcul des vitesses articulaires du robot :

Le formalisme de lagrange nécessite la connaissance de l'énergie cinétique du système physique, celle-ci nécessite aussi la connaissance de la vitesse de chaque articulation.

Soit ${}^i \mathbf{r}_i$ un point fixe (Figure 2.15) du corps i exprimé en coordonnées homogènes dans le repère attaché à l'articulation i par :

$${}^i \mathbf{r}_i = \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} = [x_i \quad y_i \quad z_i \quad 1]^T \quad (2.19)$$

Soit ${}^0 \mathbf{r}_i$ le même point exprimé dans le repère de base R_0 , et ${}^{i-1} \mathbf{T}_i$ la matrice de transformation homogène qui permet le passage du repère $i-1$ vers le repère i :

$${}^0 \mathbf{r}_i = {}^0 \mathbf{T}_i \times {}^i \mathbf{r}_i \quad (2.20)$$

$${}^0 \mathbf{T}_i = {}^0 \mathbf{T}_1 \times {}^1 \mathbf{T}_2 \times \dots \times {}^{i-1} \mathbf{T}_i \quad (2.21)$$

La vitesse de ${}^i \mathbf{r}_i$ est exprimée dans le repère de base par la relation :

$${}^0 \mathbf{v}_i = \dot{\mathbf{v}}_i = \frac{d}{dt} ({}^0 \mathbf{r}_i) = \frac{d}{dt} ({}^0 \mathbf{T}_i \times {}^i \mathbf{r}_i) \quad (2.22)$$

$$= {}^0 \dot{\mathbf{T}}_1 \times {}^1 \mathbf{T}_2 \times \dots \times {}^{i-1} \mathbf{T}_i \times {}^i \mathbf{r}_i + {}^0 \mathbf{T}_1 \times {}^1 \dot{\mathbf{T}}_2 \times \dots \times {}^{i-1} \mathbf{T}_i \times {}^i \mathbf{r}_i + \dots + {}^0 \mathbf{T}_1 \times {}^1 \mathbf{T}_2 \times \dots \times {}^{i-1} \dot{\mathbf{T}}_i \times {}^i \mathbf{r}_i + {}^0 \mathbf{T}_i \times \dot{{}^i \mathbf{r}}_i,$$

$$= \left[\sum_{j=1}^i \frac{\partial {}^0 T_i}{\partial q_j} \cdot \dot{q}_j \right] {}^i r_i \quad (2.23)$$

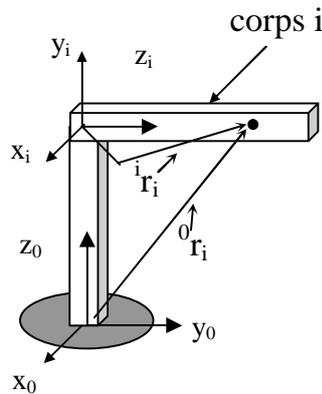


Figure 2.15 : Représentation graphique de ${}^0 r_i$ et ${}^i r_i$.

Cette formule est obtenue parce que $\dot{{}^i r_i} = 0$. La dérivé partielle de ${}^0 T_i$ par rapport à q_i peut être facilement calculée en utilisant la matrice Q_i donnée par :

- Pour une articulation prismatique : $Q_i = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix},$

- Pour une articulation rotoïde : $Q_i = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$

Ce qui permet d'écrire : $\frac{\partial {}^{i-1} T_i}{\partial q_i} = Q_i {}^{i-1} T_i \quad (2.24)$

Donc, pour $i = 1, 2, \dots, n$.

$$\frac{\partial {}^0 T_i}{\partial q_i} = \begin{cases} {}^0 T_1 {}^1 T_2 \dots {}^{j-2} T_{j-1} Q_j {}^{j-1} T_j \dots {}^{i-1} T_i & \text{pour } j \leq i \\ 0 & \text{pour } j > i \end{cases} \quad (2.25)$$

Pour simplifier la notation on définit :

$$U_{ij} = \partial {}^0 T_i / \partial q_j \quad (2.26)$$

Donc l'équation (2.25) : peut s'écrire sous la forme suivante :

$$U_{ij} = \begin{cases} {}^0T_{j-1}Q_j^{j-1}T_i & \text{pour } j \leq 1 \\ 0 & \text{pour } j > 1 \end{cases} \quad (2.27)$$

Et donc :

$$v_i = \left[\sum_{j=1}^i U_{ij} \cdot \dot{q}_j \right]^i r_i \quad (2.28)$$

Maintenant, on va calculer les effets d'interaction entre les articulations :

$$\frac{\partial U_{ij}}{\partial q_k} = U_{ijk} = \begin{cases} {}^0T_{j-1}Q_j^{j-1}T_kQ_k^{k-1}T_i & \text{pour } i \geq k \geq j \\ {}^0T_{k-1}Q_k^{k-1}T_{j-1}Q_j^{j-1}T_i & \text{pour } i \geq j \geq k \\ 0 & \text{pour } i < j \text{ ou } i < k \end{cases} \quad (2.29)$$

2.7.4.2. Calcul de l'énergie cinétique :

Après l'obtention de la vitesse articulaire de chaque corps, on a besoin de calculer l'énergie cinétique de ces corps.

Soit E_i l'énergie cinétique du corps i , $i = 1, 2, \dots, n$, et dE_i l'énergie partielle d'une masse différentielle dm du corps i , alors :

$$\begin{aligned} dE_i &= \frac{1}{2} (\dot{x}_i^2 + \dot{y}_i^2 + \dot{z}_i^2) dm \\ &= \frac{1}{2} \text{Trace}(v_i \cdot v_i^T) \cdot dm = \frac{1}{2} \text{Tr}(v_i \cdot v_i^T) \cdot dm \end{aligned} \quad (2.30)$$

En substituant v_i de l'équation I.28 dans l'équation 2.30 :

$$\begin{aligned} dE_i &= \frac{1}{2} \text{Tr} \left[\left[\sum_{p=1}^i U_{ip} \cdot \dot{q}_p \cdot {}^i r_i \right] \times \left[\sum_{r=1}^i U_{ir} \cdot \dot{q}_r \cdot {}^i r_i \right]^T \right] \cdot dm \\ &= \frac{1}{2} \left(\text{Tr} \left[\sum_{p=1}^i \sum_{r=1}^i U_{ip} ({}^i r_i \cdot dm \cdot {}^i r_i^T) \cdot U_{ir}^T \cdot \dot{q}_p \cdot \dot{q}_r \right] \right) \end{aligned} \quad (2.31)$$

En faisant l'intégral de l'énergie cinétique partielle, on obtient l'énergie cinétique du corps i :

$$E_i = \int dE_i = \frac{1}{2} \text{Tr} \left[\sum_{p=1}^i \sum_{r=1}^i U_{ip} \left(\int {}^i r_i \cdot {}^i r_i^T dm \right) \cdot U_{ir}^T \cdot \dot{q}_p \cdot \dot{q}_r \right] \quad (2.32)$$

Le terme d'intégral représente l'inertie du corps i, alors :

$$J_i = \int {}^i r_i \cdot {}^i r_i^T dm = \begin{bmatrix} \int x_i^2 dm & \int x_i y_i dm & \int x_i z_i dm & \int x_i dm \\ \int x_i y_i dm & \int y_i^2 dm & \int y_i z_i dm & \int y_i dm \\ \int x_i z_i dm & \int y_i z_i dm & \int z_i^2 dm & \int z_i dm \\ \int x_i dm & \int y_i dm & \int z_i dm & \int dm \end{bmatrix} \quad (2.33)$$

Si on utilise le tenseur d'inertie Iij qui est donné par :

$$I_{ij} = \int \left[\delta_{ij} \left[\sum_k x_k^2 \right] - x_i \cdot x_j \right] dm \quad (2.34)$$

Où δ_{ij} est appelé Kronécker delta, alors J_i peut être exprimé en forme de tenseur d'inertie comme suit :

$$J_i = \begin{bmatrix} \frac{-I_{xx} + I_{yy} + I_{zz}}{2} & I_{xy} & I_{xz} & m_i \bar{x}_i \\ I_{xy} & \frac{I_{xx} - I_{yy} + I_{zz}}{2} & I_{yz} & m_i \bar{y}_i \\ I_{xz} & I_{yz} & \frac{I_{xx} + I_{yy} - I_{zz}}{2} & m_i \bar{z}_i \\ m_i \bar{x}_i & m_i \bar{y}_i & m_i \bar{z}_i & m_i \end{bmatrix} \quad (2.35)$$

Et finalement l'énergie cinétique totale du robot est donnée par :

$$\begin{aligned} E_i &= \int dE_i = \frac{1}{2} \sum_{i=1}^n \text{Tr} \left[\sum_{p=1}^i \sum_{r=1}^i U_{ip} \cdot J_i \cdot U_{ir}^T \cdot \dot{q}_p \cdot \dot{q}_r \right] \\ &= \frac{1}{2} \sum_{i=1}^n \sum_{p=1}^i \sum_{r=1}^i \left[\text{Tr}(U_{ip} \cdot J_i \cdot U_{ir}^T) \cdot \dot{q}_p \cdot \dot{q}_r \right] \end{aligned} \quad (2.36)$$

2.7.4.3. Calcul de l'énergie potentielle :

Soit P , l'énergie potentielle totale du robot manipulateur, et P_i l'énergie potentielle de chacun de ses corps :

$$P_i = -m_i \cdot G \cdot \bar{r}_i = -m_i \cdot G \cdot ({}^0 T_i \cdot \bar{r}_i) \quad (2.37)$$

L'énergie potentielle totale est obtenue par sommation des énergies de tous les corps :

$$P = \sum_{i=1}^n P_i = \sum_{i=1}^n -m_i \cdot G \cdot ({}^0T_i \cdot \overline{{}^i r_i}) \quad (2.38)$$

où :

$G = (g_x, g_y, g_z, 0)^T$ est le vecteur colonne de gravité.

$G = (0, 0, -|g|, 0)$ et $g = 9.8062 \text{ m/sec}^2$ est la constante de gravité.

2.7.4.4. Equation de mouvement du robot manipulateur :

A partir des équations (I.36) et (I.38), le lagrangien $L = E - P$ est donné par :

$$L = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^i \sum_{k=1}^i \left[\text{Tr}(U_{ij} \cdot J_i \cdot U_{ij}^T) \cdot \dot{q}_j \cdot \dot{q}_k \right] + \sum_{i=1}^n m_i \cdot g \cdot ({}^0T_i \cdot \overline{{}^i r_i}) \quad (2.39)$$

Appliquant le formalisme de Lagrange sur L :

$$\begin{aligned} \Gamma_i &= \frac{d}{dt} \left[\frac{\partial L}{\partial \dot{q}_i} \right] - \frac{\partial L}{\partial q_i}, \\ &= \sum_{j=i}^n \sum_{k=1}^j \text{Tr}(U_{jk} \cdot J_j \cdot U_{ji}^T) \cdot \ddot{q}_k + \sum_{j=i}^n \sum_{k=1}^j \sum_{m=1}^j \text{Tr}(U_{jkm} \cdot J_j \cdot U_{ji}^T) \cdot \dot{q}_k \cdot \dot{q}_m \\ &\quad - \sum_{j=i}^n m_j \cdot G \cdot U_{ji} \cdot \overline{{}^j r_j} \end{aligned} \quad (2.40)$$

L'équation ci-dessus peut être exprimée dans une notation plus simple comme suit :

$$\Gamma_i = \sum_{k=1}^n D_{ik} \cdot \ddot{q}_k + \sum_{k=1}^n \sum_{m=1}^n h_{ikm} \cdot \dot{q}_k \cdot \dot{q}_m + C_i \quad (2.41)$$

où sous forme matricielle comme suit :

$$\Gamma(t) = D(q(t)) \cdot \ddot{q}(t) + h(q(t), \dot{q}(t)) + C(q(t)) \quad (2.42)$$

Où :

$\Gamma(t)$: $n \times 1$ vecteur des couples appliqués aux joints du robot :

$$\Gamma(t) = (\Gamma_1(t), \Gamma_2(t), \dots, \Gamma_n(t)) \quad (2.43)$$

$q(t)$: $n \times 1$ vecteur des positions articulaires :

$$q(t) = (q_1(t), q_2(t), \dots, q_n(t)) \quad (2.44)$$

$\dot{q}(t)$: $n \times 1$ vecteur des vitesses articulaires :

$$\dot{q}(t) = (\dot{q}_1(t), \dot{q}_2(t), \dots, \dot{q}_n(t)) \quad (2.45)$$

$\ddot{\mathbf{q}}(t)$: $n \times 1$ vecteur des accélérations articulaires :

$$\ddot{\mathbf{q}}(t) = (\ddot{q}_1(t), \ddot{q}_2(t), \dots, \ddot{q}_n(t)) \quad (2.46)$$

$D(\mathbf{q})$: $n \times n$ matrice d'inertie, ses éléments sont donnés par :

$$D_{ik} = \sum_{j=\max(i,k)}^n \text{Tr}(\mathbf{U}_{jk} \cdot \mathbf{J}_j \cdot \mathbf{U}_{ji}^T) \quad i, k = 1, 2, \dots, n \quad (2.47)$$

$\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}})$: $n \times 1$ vecteur englobant les forces de coriolis et de centrifuges :

$$\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) = (h_1, h_2, \dots, h_n)^T$$

où :
$$h_i = \sum_{k=1}^n \sum_{m=1}^n h_{ikm} \cdot \dot{q}_k \cdot \dot{q}_m \quad (2.48)$$

et :
$$h_{ikm} = \sum_{j=\max(i,k,m)}^n \text{Tr}(\mathbf{U}_{jkm} \cdot \mathbf{J}_j \cdot \mathbf{U}_{ji}^T) \quad i, k, m = 1, 2, \dots, n \quad (2.49)$$

$C(\mathbf{q})$: $n \times 1$ vecteur des couples dus aux forces de gravité :

$$C(\mathbf{q}) = (C_1, C_2, \dots, C_n)$$

où :
$$C_i = \sum_{j=1}^n (-m_i \cdot \mathbf{g} \cdot \mathbf{U}_{ji} \cdot {}^j \mathbf{r}_j) \quad i = 1, 2, \dots, n \quad (2.50)$$

L'équation 2.42 représente le modèle dynamique du robot manipulateur qui peut être utilisé dans les applications de simulation.

Chapitre 3

Les algorithmes génétiques

3. Les algorithmes génétiques.

3.1. Introduction :

Le désir humain de perfection trouve son expression dans la théorie de l'optimisation qui étudie comment décrire et atteindre ce qui est meilleur, une fois que l'on connaît comment mesurer et modifier ce qui est bon et mauvais.

Dans la pratique un grand nombre de fonctions à optimiser ne sont pas dérivables et souvent même pas continues, ainsi que le monde réel à explorer est envahi de discontinuités, d'espaces de recherches bruités et multi-modaux, ce qui le rend moins adapté au calcul. Il n'est donc pas surprenant que les méthodes d'optimisation soumises aux contraintes de continuité et de dérivabilité ne soient adaptées qu'à une classe de problèmes très limitée, pour cette raison, et à cause du caractère intrinsèque local de leur exploration, nous devons rejeter ces méthodes classiques. Récemment, une nouvelle classe de méthodes d'optimisation est apparue employant les principes d'évolution et d'hérédité de la nature et présentant une probabilité importante de convergence vers un optimum global de la fonction à optimiser, ce sont des méthodes pseudo-aléatoires appelées "LES ALGORITHMES GENETIQUES".

Les algorithmes génétiques (AGs) ont été développés par John Holland à l'université du Michigan, ses recherches avaient comme objectif de [48] :

- Mettre en évidence et expliquer rigoureusement les processus d'adaptation des systèmes naturels.
- Concevoir des systèmes artificiels possédant des propriétés importantes des systèmes naturels.

Les algorithmes génétiques sont des algorithmes d'exploration fondés sur les mécanismes de la sélection naturelle et de la génétique. Ils utilisent à la fois les principes de la survie des structures les mieux adaptées, et les échanges d'information pseudo-aléatoires, pour former un algorithme d'exploration qui possède certaines caractéristiques de l'exploration

humaine. A chaque génération, un nouvel ensemble de créatures artificielles (des chaînes de caractères) est créé en utilisant des parties des meilleurs éléments de la génération précédente. Bien qu'utilisant le hasard, les algorithmes génétiques ne sont pas purement aléatoires, ils explorent efficacement l'information obtenue précédemment pour spéculer sur la position de nouveaux points à explorer, avec l'espoir d'améliorer la performance.

3.2. Principe de fonctionnement des AGs :

Les AGs effectuent le procédé d'optimisation en agissant sur une population de créatures artificielles (chaînes de caractères) analogues aux chromosomes en nature. Chaque créature ou individu représente un point de recherche dans l'espace des solutions à qui on associe une valeur de fonction coût dont on veut obtenir la valeur maximale.

Une description abstraite d'un AG de base peut se faire selon les étapes suivantes[48] :

1. Initialiser aléatoirement une population de chromosomes (individus).
2. Evaluer chaque chromosome dans la population, à chaque chromosome est associé une fonction coût ou fonction d'aptitude déterminant son rang dans la population. Cette fonction est l'arbitre final décidant la vie ou la mort de chaque individu.
3. Créer de nouveaux chromosomes en appliquant la sélection et les opérateurs de reproduction.
4. Evaluer les nouveaux chromosomes (les descendants) et les insérer dans la population pour construire une nouvelle génération.

Ce processus se répète jusqu'à la satisfaction du critère d'arrêt de l'AG qui est généralement spécifié par un nombre maximal de générations. L'organigramme suivant donne une description générale de l'AG standard [49].

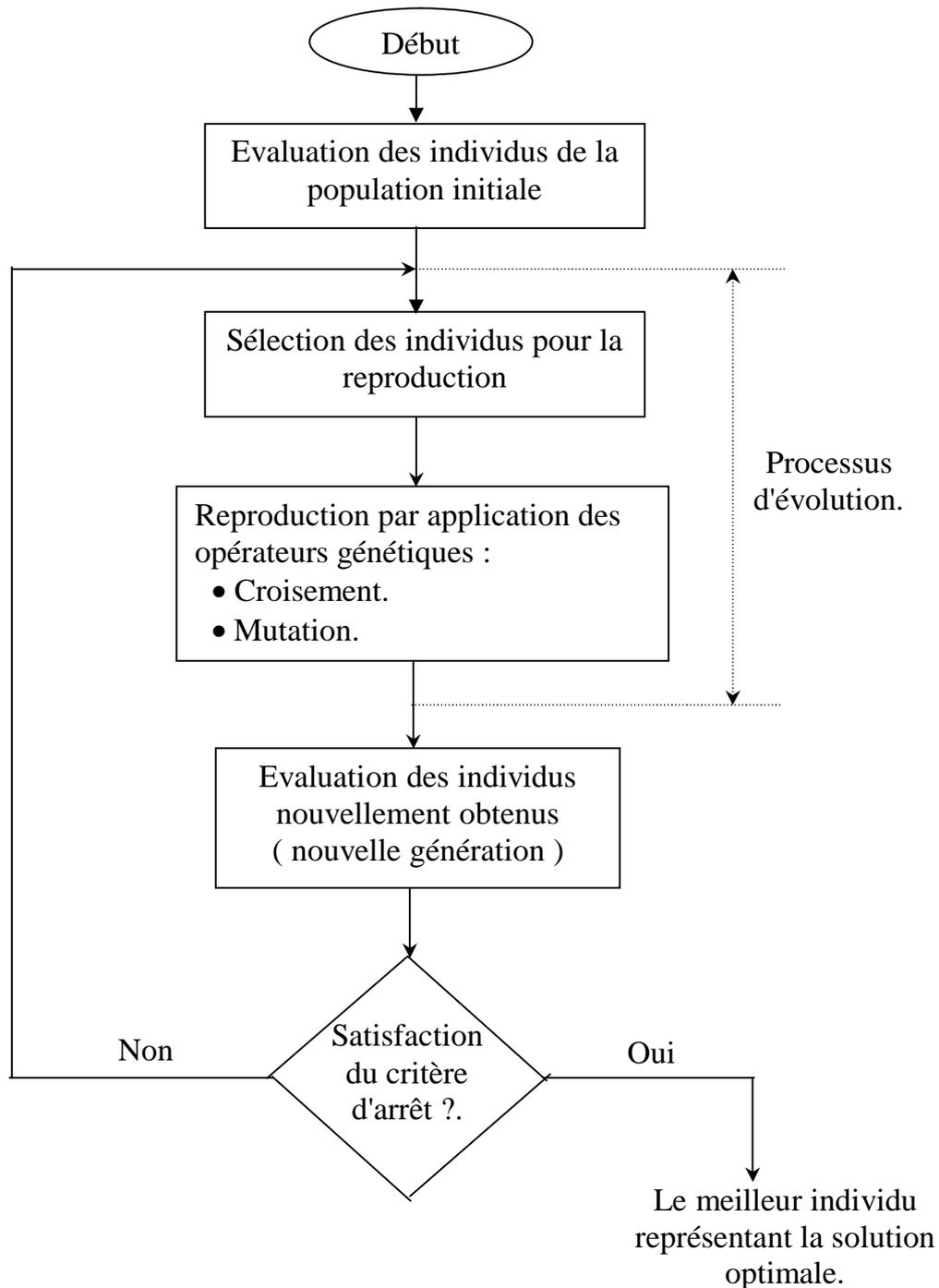


Figure 3.1: Organigramme d'un AG standard.

3.3. Ce qui rend les algorithmes génétiques différents des méthodes classiques ? :

Pour qu'ils puissent surpasser leurs cousins plus classiques dans la quête de la robustesse, les algorithmes génétiques doivent être fondamentalement différents. Ils le sont en fait selon quatre axes principaux [48] :

1. Les algorithmes génétiques utilisent un codage des paramètres, et non pas les paramètres eux-mêmes.
2. Les algorithmes génétiques travaillent sur une population de points, au lieu d'un point unique.
3. Les algorithmes génétiques n'utilisent que les valeurs de la fonction étudiée, pas sa dérivée ou une autre connaissance auxiliaire.
4. Les algorithmes génétiques utilisent des règles de transition probabilistes, et non déterministes.

3.4. Formulation du problème d'optimisation :

En raison de l'analogie avec la théorie de l'évolution naturelle (survie des individus les mieux adaptés à leur environnement), l'algorithme génétique est naturellement formulé en terme de maximisation. Etant donnée une fonction f réelle à une ou plusieurs variables, le problème d'optimisation sur l'espace de recherche E s'écrit de la manière suivante :

$$\max_{x \in E} f(x) \quad (3.1)$$

De plus, la fonction à optimiser par un algorithme génétique doit avoir des valeurs positives sur l'ensemble du domaine E . Dans le cas contraire, il convient d'ajouter aux valeurs de f une constante positive f_{\min} :

$$\max_{x \in E} f(x) + f_{\min} \quad (3.2)$$

Dans beaucoup de problèmes, l'objectif est exprimé sous la forme de minimisation d'une fonction coût g donnée par :

$$\min_{x \in E} g(x) \quad (3.3)$$

Le passage du problème de minimisation à un problème de maximisation est obtenu par transformation de la fonction g . La transformation souvent utilisée est :

$$\max_{x \in E} h(x) \quad (3.4)$$

Avec :

$$h(x) = \begin{cases} G_{\max} - g(x) & \text{si } g(x) \leq G_{\max} \\ 0 & \text{si non} \end{cases} \quad (3.5)$$

La fonction $h(x)$ n'est pas unique, en effet, toute composition de la fonction g par une fonction quelconque conduit à un problème de maximisation équivalent à (3.3). On rencontre notamment dans la littérature la fonction de transformation suivante :

$$h(x) = \frac{1}{1+g(x)} \quad (3.6)$$

3.5. Les mécanismes d'un AG standard :

Les mécanismes d'un algorithme génétique de base sont étonnamment simples et ne mettent en jeu rien de plus compliqué que des copies de chaînes et des échanges de morceaux de chaînes. Les raisons pour lesquelles ce procédé simple fonctionne sont bien plus complexes et subtiles. La simplicité de mise en œuvre et l'efficacité constituent deux des caractéristiques les plus attrayantes de l'approche proposée par les algorithmes génétiques.

Un algorithme simple composé des opérateurs suivants : la sélection, le croisement et la mutation, conduit généralement à de bons résultats pour un grand nombre d'applications.

3.5.1. La sélection :

La sélection est un procédé dans lequel chaque chaîne est copiée en fonction des valeurs de la fonction à optimiser f (cette fonction est appelée aussi la fonction d'adaptation). Intuitivement cette fonction peut être envisagée comme une mesure de profit, d'utilité ou de qualité, que l'on souhaite maximiser.

Copier des chaînes en fonction des valeurs de leurs fonctions d'adaptation revient à donner aux chaînes dont la valeur est plus grande une probabilité plus élevée de contribuer à la génération suivante, en créant au moins un descendant. Cet opérateur est bien entendu une version artificielle de la sélection naturelle, la survie darwinienne des chaînes les plus adaptées. Dans les populations naturelles, l'adaptation est déterminée par la capacité d'une créature à survivre aux prédateurs, aux maladies et aux autres obstacles à franchir pour atteindre l'âge adulte et la période de reproduction.

Dans notre environnement indéniablement artificiel, la fonction à optimiser est l'arbitre final décidant la vie ou la mort de chaque chaîne créature.

L'opérateur de sélection peut être mis en œuvre sous forme algorithmique de différentes façons :

3.5.1.1. La sélection proportionnelle (roue de loterie) :

Le mode de sélection des parents consiste à dupliquer chaque individu de la population proportionnellement à son adaptation dans son milieu.

Soit :

f_j : la valeur de la fonction coût associée au $j^{\text{ème}}$ individu,

f_s : la somme des valeurs de cette fonction,

La sélection pouvant être faite en utilisant le rapport (f_j/f_s) pour réaliser une roulette pondérée ou chaque individu occupe une surface proportionnelle au rapport précédent, des tirages aléatoires sur cette roulette donneront les chaînes qui participeront à la prochaine génération. De cette façon, les chaînes bien adaptées ont un grand nombre de descendants dans les générations suivantes.

3.5.1.2. La sélection à reste stochastique :

Dans ce mode de sélection, le nombre de copies $n(a_j^t)$ d'un individu a_j^t est directement fixé par le rapport (f_j/f_{moy}) où f_{moy} est la valeur moyenne de la fonction coût. Dans un premier temps, on reproduit chaque individu (partie entière de (f_j/f_{moy})) fois :

$$n(a_j^t) = \text{partie entière de } [f_j/f_{\text{moy}}] \quad (3.7)$$

Ensuite la population est complétée par tirage au sort en associant à chaque individu a_j^t une probabilité $P_s(a_j^t)$:

$$p_s(a_j^t) = f_j/f_{\text{moy}} - \text{partie entière de } [f_j/f_{\text{moy}}] \quad (3.8)$$

3.5.1.3. La sélection par tournoi stochastique :

Dans cette méthode de sélection, les probabilités de sélection sont calculées normalement et des paires successives d'individus sont tirées au sort grâce à la sélection par roue de loterie. Après avoir tiré ces paires de chaînes, la chaîne ayant l'adaptation la plus élevée est déclarée vainqueur, elle est ajoutée à la nouvelle population, et une nouvelle paire est tirée.

Ce processus continue jusqu'à ce que la population soit remplie.

3.5.2. Le croisement :

Après la sélection, une opération de croisement peut avoir lieu en deux étapes. Premièrement, les éléments nouvellement produits par la sélection sont appariés. Deuxièmement, chaque paire de chaînes subit un croisement comme suit : un entier k représentant une position sur la chaîne est choisi aléatoirement entre 1 et la longueur de la chaîne moins 1 (entre 1 et $L-1$). Deux nouvelles chaînes sont créées en échangeant tous les caractères compris entre les positions $k+1$ et L incluse.

Par exemple, considérons les chaînes parent₁ et parent₂ de la population initiale:

parent₁ : 1 0 1 1 1 1 0 1

parent₂ : 1 0 0 0 0 0 1 0

Supposons qu'en choisissant au hasard un nombre entre 1 et 7, nous obtenions k = 3. Le croisement conduit à la production de deux nouvelles chaînes enfant₁ et enfant₂ comme le montre la figure 3.2.

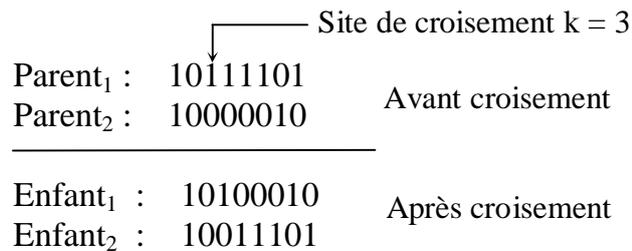


Figure 3.2: Croisement simple.

3.5.3. La mutation :

Dans un algorithme génétique de base, la mutation est la modification aléatoire occasionnelle (de faible probabilité) de la valeur d'un caractère de la chaîne, elle est considérée comme une police d'assurance protégeant de la perte prématurée de notions importantes [48].

Par exemple, considérons la chaîne A de la population initiale :

A : 10111111,

Supposons qu'en choisissant au hasard un nombre entre 1 et L-1, nous obtenions k = 4, la mutation conduit à la production de la nouvelle chaîne A' (figure 3.3).

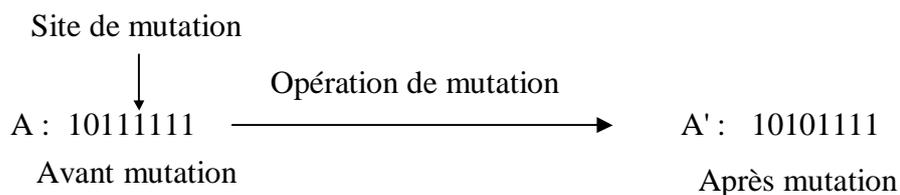


Figure 3.3: Opération de mutation

Du rôle secondaire de l'opérateur de mutation dans les AGs, la fréquence de mutation conseillée pour obtenir de bons résultats se situe autour d'une mutation tous les 1000 bits (positions). [50] préconise plutôt la valeur suivante pour p_m :

$$P_m = \frac{1}{L} \quad (3.9)$$

Où L est la longueur du chromosome.

D'autres études ont conduit à une formule empirique qui exprime le taux optimal de mutation en fonction de la longueur du chromosome L et de la taille de la population N sous la forme suivante :

$$P_m = \frac{1}{N \cdot \sqrt{L}} \quad (3.10)$$

3.6. Notion du codage :

L'utilisation des AGs nécessite, en première étape, le codage de l'ensemble des paramètres à optimiser en une chaîne de caractères de longueur finie, cette longueur est fixée par l'utilisateur selon :

- Le domaine de variation de la chaîne,
- La précision demandée.

Les possibilités de codages sont nombreuses, on peut citer, entre autres, le codage binaire, le codage gray et le codage en virgule flottante si les paramètres à optimiser ont des valeurs réelles. Bien que ces codages aient donné une certaine flexibilité, ils ne constituent pas une méthode pratique pour le codage des fonctions multi-variables $f(x_1, x_2, \dots, x_n)$. L'une des méthodes de codage proposées, qui a été appliquée avec succès, est le codage concaténé et multi-paramétré qui se déroule de la façon suivante :

- Coder chaque variable x_i ($i = 1, 2, \dots, n$) selon le choix de sa longueur.
- Construire les chaînes en concaténant les différents codes, c'est à dire, juxtaposer ces derniers les uns après les autres pour construire les chaînes (chromosomes).

Exemple:

$$\text{Chaîne : } (\text{code}_{x_1})(\text{code}_{x_2}) \dots (\text{code}_{x_n}).$$

3.7. Notion du décodage :

C'est l'opération inverse de la précédente, chaque chaîne doit être décodée pour pouvoir calculer la valeur de la fonction d'aptitude qui lui est associée. Parmi les types de décodage existant, le décodage binaire est le plus utilisé :

$$X = U_{\min} + D(100\dots 1_B) \cdot \frac{U_{\max} - U_{\min}}{2^L - 1} \quad (3.11)$$

Avec U_{\max} , U_{\min} sont les bornes supérieure et inférieure du domaine de variation de la variable X , $D(100\dots 1_B)$ est la valeur décimale convertie du binaire et L est la longueur de la chaîne codant la variable X . Le choix de la longueur dépend essentiellement de la précision voulue sur la variable X qui est donnée par :

$$\text{Précision} = \frac{U_{\max} - U_{\min}}{2^L - 1} \quad (3.12)$$

3.8. Théorie fondamentale des AGs (notion de schème) :

Un schème est un motif de similarité décrivant un sous-ensemble de chaînes avec des similarités à des positions définies, il permet une évaluation plus rigoureuse des performances des algorithmes génétiques. Un schème est représenté par une chaîne définie sur un alphabet étendu $\{0, 1, *\}$ pour lequel le "0" et le "1" gardent la même signification et le "*" est le symbole indifférent. Cette notation simplifie énormément l'analyse des AGs, car elle permet de reconnaître explicitement l'ensemble des similarités possibles dans une population de chaînes.

Exemples :

Le schème $S = (*110*)$ décrit l'ensemble de quatre éléments $[01100, 01101, 11100, 11101]$.

Deux caractéristiques importantes des schèmes sont utilisées pour décrire et classer les similarités entre les chaînes :

- L'ordre : l'ordre d'un schème S noté $O(S)$, est le nombre de positions instanciées (nombre de 1 et de 0 pour un alphabet binaire) dans le schème considéré.

Par exemple : $O(*111*) = 3$.

- La longueur utile : la longueur utile d'un schème S noté $\delta(S)$ est la distance entre la première et la dernière positions instanciées dans le schème considéré.

Par exemple : $\delta(*111*) = 4 - 2 = 2$.

Après avoir défini les schèmes et leurs caractéristiques, on passe à l'analyse de développement et de disparition des schèmes importants de génération en génération, en analysant l'effet des différents opérateurs génétiques sur les schèmes contenus dans la population.

Considérons une population de taille N contenant à l'instant t m exemplaires d'un schème S noté $m(S, t)$.

3.8.1. Effet de la sélection :

Supposons que la sélection est de type proportionnel, son effet sur un schème est facile à déterminer, puisque les chaînes les mieux adaptées ont une plus grande probabilité de sélection, on choisit globalement sans cesse davantage les schèmes qui semblent les meilleurs.

Lors de la sélection, les chaînes représentant le schème S sont sélectionnées avec une probabilité moyenne $f_{\text{moy}} = \frac{\sum f_i}{N}$, où f_i est la fonction d'objectif de la chaîne i, donc le nombre d'exemplaires attendus d'un schème S à l'instant t+1 est donnée par :

$$m(S, t+1) = m(S, t) \frac{f(S)}{f_{\text{moy}}} \quad (3.13)$$

Où $f(S)$ et f_{moy} sont respectivement, l'aptitude moyenne des individus représentés par le schème S à la génération t et l'aptitude moyenne de la population.

L'expression (3.13) montre que les schèmes ayant une aptitude supérieure à la moyenne de la population auront plusieurs copies dans la nouvelle génération, par contre les schèmes dont les valeurs d'aptitude sont inférieures à la moyenne auront moins de copies. Donc deux cas se présentent : un développement dans le premier cas et un dépérissement dans le deuxième.

Pour démontrer le processus de développement et de dépérissement, on suppose qu'un schème S reste au-dessus de la moyenne avec une quantité égale à $c.f_{\text{moy}}$, où c est une constante. Ceci permet de réécrire l'équation (3.13) sous une nouvelle forme :

$$m(S, t+1) = m(S, t) \cdot (f_{\text{moy}} + c.f_{\text{moy}}) / f_{\text{moy}} = (1+c) \cdot m(S, t) \quad (3.14)$$

Si l'aptitude d'un schème quelconque reste au-dessous de la moyenne de la population par une certaine quantité c, le taux de développement ou de dépérissement d'un schème prend une forme exponentielle, donc l'équation (3.14) devient :

$$m(S, t+1) = (1+c)^t \cdot m(S, 0) \quad (3.15)$$

3.8.2. Effet du croisement :

L'effet de l'opérateur de croisement sur un schème particulier dépend de sa longueur utile. Un schème survit si le point de croisement tombe à l'extérieur de sa longueur utile, donc plus la longueur utile est courte plus la probabilité de survie est élevée.

La probabilité de destruction d'un schème S de longueur utile $\delta(S)$ lors d'un croisement simple est donnée par :

$$p_d(S) = \frac{\delta(S)}{L-1} \quad (3.16)$$

où L est la longueur du chromosome.

Si le croisement a lieu avec une probabilité p_c , la probabilité de survie est alors :

$$p_s(S) \geq 1 - p_c \frac{\delta(S)}{L-1} \quad (3.17)$$

L'effet combiné des opérations de sélection et de croisement est obtenu en multipliant le nombre attendu des schèmes lors de la sélection seule par la probabilité p_s de survie lors d'un croisement, on obtient alors :

$$m(S, t+1) \geq m(S, t) \frac{f(S)}{f_{\text{moy}}} \left[1 - p_c \frac{\delta(S)}{L-1} \right] \quad (3.18)$$

Il est clair que les schèmes de longueurs $\delta(S)$ courtes et d'aptitudes $f(S) > f_{\text{moy}}$ auront un nombre de copies plus élevé dans la nouvelle génération $t+1$.

3.8.3. Effet de la mutation :

La mutation est la modification aléatoire d'une position d'un chromosome avec une probabilité p_m souvent petite. Un schème survit si toutes ses positions instanciées restent inchangées, si la probabilité de survie d'une position quelconque est $(1 - p_m)$, et le nombre de positions fixes dans le schème est $O(S)$, la probabilité de survie d'un schème S à la mutation est :

$$p_s(S) = (1 - p_m)^{O(S)} \quad (3.19)$$

Et puisque $p_m \ll 1$, l'expression (II.19) peut être approximée par la relation :

$$p_s \approx 1 - O(S) \cdot p_m \quad (3.20)$$

3.8.4. Effet combiné :

Nous pouvons donc conclure qu'on peut s'attendre à ce qu'un schème S quelconque reçoive, du fait de la sélection, du croisement et de la mutation, un nombre de copies donné par l'équation suivante :

$$m(S, t+1) \geq m(S, t) \frac{f(S)}{f_{\text{moy}}} \left[1 - O(S) \cdot p_m - p_c \frac{\delta(S)}{L-1} \right] \quad (3.21)$$

Ce qui montre clairement que les schèmes de longueurs utiles courtes, d'ordres faibles et de performances au-dessus de la moyenne font l'objet d'un nombre d'exemplaires

exponentiellement croissant dans les générations suivantes. Cette conclusion est importante, d'une importance telle qu'elle est appelée la théorie fondamentale des algorithmes génétiques [48].

Les opérateurs génétiques à savoir la sélection, le croisement et la mutation sont généralement suffisants pour accomplir la tâche d'optimisation. L'AG dans ce cas sera considéré comme une boîte noire d'optimisation de fonctions ne nécessitant aucune connaissance sur le domaine. Cependant, l'amélioration de l'AG standard et éventuellement la connaissance du domaine peuvent être introduites et exploitées pour améliorer les performances de l'AG.

3.9. Amélioration des techniques de base des AGs :

Les algorithmes génétiques ont démontré leur efficacité pour la résolution d'une large classe de problèmes difficiles et pour la recherche dans des espaces complexes, leur efficacité et leur performance peuvent être renforcées et leur utilisation peut être étendue à de nouvelles classes de problèmes en introduisant des modifications et des extensions au niveau de leurs stratégies et opérateurs génétiques de base. Les efforts de recherche consacrés dans cette direction peuvent être résumés dans les points suivants:

3.9.1. Amélioration des opérateurs de reproduction :

3.9.1.1. La sélection :

Plusieurs stratégies sont possibles :

a. La sélection par descendance :

Les individus de la nouvelle génération sont obtenus par descendance où les enfants remplacent automatiquement leurs parents quelle que soit leur adaptation.

L'inconvénient de ce mode de sélection est que l'on risque de voir disparaître les caractéristiques des parents les mieux adaptés si elles n'ont pas été totalement transmises lors de la recombinaison génétique.

b. La sélection par compétition :

Une compétition a lieu entre parents et enfants pour déterminer les survivants de la génération. Ainsi, les enfants peuvent être insérés dans la population si et seulement si leur performance est supérieure à celle de leurs parents à rang équivalent.

$$a_i^{t+1} = \begin{cases} a_i^{''t} & \text{si } f(a_i^{''t}) > f(a_i^{'t}) \\ a_i^{'t} & \text{sinon} \end{cases} \quad (3.22)$$

Avec a_i^{t+1} est le $i^{\text{ème}}$ individu de la génération ($t+1$), $a_i^{'t}$ et $a_i^{''t}$ représentent respectivement le parent (i) et l'enfant (i) de la génération (t).

c. Sélection en régime permanent :

Cette technique consiste à effectuer une compétition après chaque recombinaison génétique entre parents et enfants en retenant les deux meilleurs individus parmi les quatre.

d. Sélection par procréation sélective :

Dans cette méthode, on garde les N meilleurs individus parmi la population intermédiaire de parents et d'enfants pour former la nouvelle génération. Ceci garantit que les meilleurs individus de la population (qu'ils soient enfants ou parents) seront nécessairement conservés pour la génération suivante.

3.9.1.2. L'opérateur de croisement :

a. Opérateur de croisement en deux points :

Le principe de cette opération, comme on peut le voir sur la figure (3.4), est le même que celui en un seul point, seulement au lieu de choisir un seul point de croisement, on doit choisir deux points et on échange l'ensemble de caractères existants entre ces deux points.

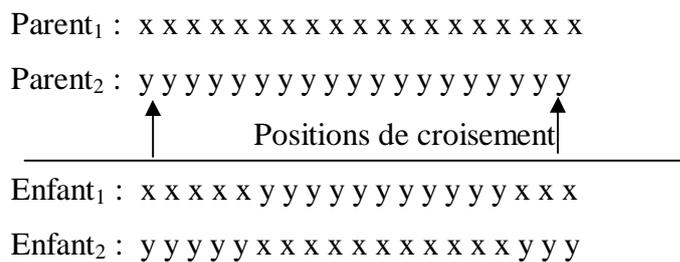


Figure 3.4: Opérateur de croisement en deux points

b. Opérateur de croisement aléatoire :

Cet opérateur combine deux chromosomes selon une chaîne binaire aléatoire. A chaque position de cette chaîne aléatoire, les caractères correspondants sont échangés si et

3.9.2.2. La troncature en sigma :

Dans cette procédure, on utilise l'écart type de la population, ce qui permet d'éliminer les individus trop faibles.

La nouvelle fonction coût s'écrit sous la forme :

$$f' = f - (f_{\text{moy}} - c \cdot \sigma) \quad (3.24)$$

où :

c : Coefficient de l'écart type choisi arbitrairement $1 \leq c \leq 3$.

σ : L'écart type de l'adaptation calculée sur la population.

3.9.2.3. Le changement d'échelle en puissance :

Dans ce cas, l'adaptation transformée f' prend la valeur d'une certaine puissance de l'adaptation brute f :

$$f' = f^k \quad (3.25)$$

Où la constante k peut être adaptée au cours des générations pour accentuer la sélection des individus situés à proximité des sommets.

3.9.3 Optimisation des paramètres de l'algorithme génétique :

Les principaux paramètres d'un algorithme génétique sont [49] : la probabilité de mutation p_m , la probabilité de croisement p_c et la taille de la population N , le choix de ces paramètres joue un rôle très important dans l'emploi des AGs comme procédure d'optimisation. Si par exemple, la taille de la population est choisie petite, cela implique une exécution rapide mais une convergence incertaine, si la taille est importante, le problème du temps de calcul apparaît. La sélection des meilleurs individus pour la reproduction et la diversité dans la population sont deux facteurs très importants pour le déroulement de l'AG dans un sens positif, ces deux facteurs sont liés directement à la taille de la population et aux valeurs de p_m et p_c .

L'idée d'autorégulation des paramètres de l'AG a été développée par plusieurs chercheurs, cette idée peut être schématisée comme suit :

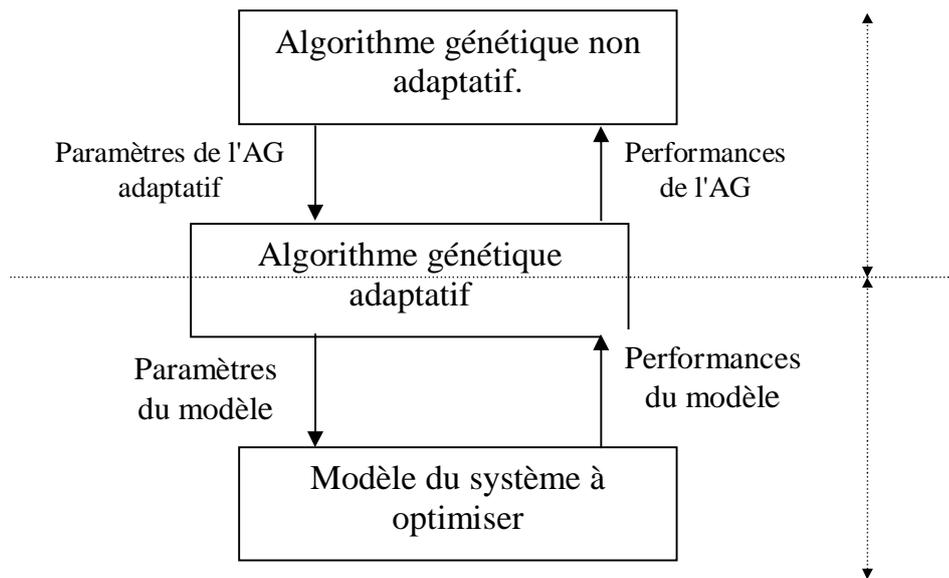


Figure 3.6: Algorithme génétique adaptatif

De nombreuses études ont été faites sur l'effet d'adaptation des paramètres de l'AG sur son déroulement [49], le résultat de ces études est la formulation d'une relation permettant d'avoir des paramètres optimaux de l'AG, cette relation est donnée par :

$$\ln(N) + 0.93 \ln(p_m) + 0.456 \ln(L) = 0.56 \quad (3.26)$$

Où, L est la longueur du chromosome, cette relation peut être approximée par :

$$N \cdot p_m \cdot \sqrt{L} = 1.7 \quad (3.27)$$

Cette formule montre que la probabilité de mutation est inversement proportionnelle à la taille de la population.

3.9.4. Introduction des contraintes :

Les AGs sont par leur nature des techniques d'explorations non-containtes, cependant, la plupart des problèmes réels contiennent une ou plusieurs contraintes qui doivent être satisfaites, ces contraintes sont d'habitude exprimées sous forme d'égalités ou d'inégalités. Les premières peuvent être introduites dans le modèle du système, mais pour les deuxièmes on procède comme suit :

Pour chaque chromosome nouvellement produit, on évalue la fonction d'objectif et on vérifie qu'aucune des contraintes n'est violée. Si tel est le cas, la fonction d'objectif gardera sa valeur. Si certaines contraintes sont violées, la solution est rejetée et la fonction d'objectif n'a pas donc de valeur.

L'obtention d'une solution acceptable en appliquant cette procédure sur des problèmes très contraints devient très difficile, c'est pourquoi on fait appel souvent aux méthodes de pénalisation qui transforment le problème d'optimisation contraint en un problème non-contraint, ceci en associant un coût ou une pénalité à chaque violation de contrainte. Ce coût est inclus dans l'évaluation de la fonction d'objectif pour refléter à la fois la qualité et l'acceptabilité des solutions [48].

3.10. Conclusion :

Les AGs sont des outils d'optimisation très puissants, ils explorent l'espace de recherche par une population et non pas par un point unique, ce qui réduit la probabilité d'atteindre un faux pic.

Les AGs réalisent une optimisation efficace en ne tenant compte que des valeurs de la fonction à optimiser. Les autres méthodes utilisent beaucoup d'informations auxiliaires, si bien que dans les problèmes pour lesquels cette information n'est pas disponible ou difficile à obtenir, ces techniques sont bloquées. Les AGs restent généraux en utilisant que l'information disponible dans tous les problèmes d'optimisation. Les AGs traitent simultanément les similarités dans le codage sous-jacent et l'information qui permet d'ordonner les structures en fonction de leur capacité de survie dans l'environnement considéré. En exploitant un type d'information si généralement disponible, les AGs peuvent être appliqués virtuellement à tous les problèmes.

Chapitre 4

La logique floue

4.1. Histoire de Logique Floue

La théorie de la logique floue a été développée dans les années soixante à l'université de Berkeley par le professeur Lotfi Zadeh. Il travaillait à Berkeley depuis 1959 et il a été président de *l'Electrical Engineering department* de 1963 à 1968. Le texte *Fuzzy sets* est apparu la première fois en 1965 [51] dans la revue *Information and Control*. Il est considéré généralement comme le début de la théorie, bien qu'une ébauche en ait été donnée plus tôt dans l'histoire, par Lukasiewicz en Pologne dans les années vingt. Depuis 1965, le professeur Zadeh n'a pas arrêté d'enrichir la théorie dans ce domaine avec ses innombrables publications dans différents journaux et conférences. Mais ce n'est que depuis les années quatre vingt, que les idées formulées dans les années soixante ont été acceptées. Dans ses écrits, le professeur L. Zadeh raconte comment il a été marginalisé à cause de ses nouvelles idées qu'il véhiculait. Tels n'était pas le cas au Japon, où la théorie de la logique floue a reçu une très grande attention. Des sommes considérables ont fait l'objet de donations pour des fins de recherches dans ce nouveau domaine, ce qui a abouti à de nombreux produits utilisant cette technique. Cependant, la première recherche pratique sur cette nouvelle théorie a eu lieu au Queen Mary College de Londres. En effet, en 1974, Mamdani et ses étudiants motivés par les articles de Zadeh, ont réussi à établir le premier rapprochement entre la théorie du flou et le contrôle des systèmes. Il s'agissait de la régulation d'une machine à vapeur. Ce succès surprenant a motivé par la suite King et Mamdani [52] au contrôle de la température d'un réacteur chimique. Cependant, la première application industrielle était en 1980 destinée à la conduite d'un four à chaux de F.L.Smith & Co A/S au Danemark. Une autre application pratique de cette technique a été réalisée en 1983 au Japon pour la conduite d'installation d'assainissement d'eau.

Parmi les applications de la logique floue est le réglage et la commande des manipulateurs mobiles. La commande d'un système lorsque des imperfections entachent la connaissance dont nous disposons sur ce système, lorsqu'une modélisation rigoureuse de celui-ci est difficile et lorsqu'il est très complexe, demeurerait une tâche très délicate jusqu'à l'apparition de la notion de la logique floue qui a changé radicalement la philosophie de la commande. Ceci est dû au fait que les lois de commande conventionnelles sont remplacées par des variables linguistiques dans des relations conditionnelles floues.

Le terme de "logique floue" a deux acceptions [53] :

- La première, correspond à tout le développement concernant la théorie des ensembles flous,
- La deuxième, représente une extension de la logique classique, dans le but de raisonner sur des connaissances imparfaites.

4.2. Théorie des ensembles flous :

4.2.1. Concepts fondamentaux flous :

Le concept d'ensemble flou a été introduit pour éviter le passage brusque d'une classe à une autre (par exemple, de la classe noire à la classe blanche) et autoriser des éléments à n'appartenir complètement ni à l'une ni à l'autre (à être gris, par exemple), ou encore à appartenir partiellement à chacune (avec un fort degré à la classe noire et un faible degré à la classe blanche, dans le cas du gris foncé). La définition d'un ensemble flou répond au besoin de représenter des connaissances imprécises, soit parce qu'elles sont exprimées en langage naturel par un observateur qui n'éprouve pas le besoin de fournir plus de précision ou n'en est pas capable, soit parce qu'elles sont obtenues avec des instruments d'observation qui produisent des erreurs de mesure [54].

La notion des ensembles flous nous permet de traiter :

- Des catégories aux limites mal définies.
- Des situations intermédiaires entre le tout et le rien.
- Le passage progressif d'une propriété à une autre.
- Des valeurs approximatives.

Donc, le concept des ensembles flous constitue un assouplissement de celui d'un ensemble donné.

4.2.2. Définition d'un ensemble flou :

Soit le référentiel ou l'univers de discours U qui est le domaine de définition d'une variable x , un ensemble A dans U est défini par l'ensemble des paires ordonnées (élément générique de U , degré d'appartenance) :

$$A = \{ (x, \mu_A(x)) \mid x \in U \} \quad (4.1)$$

Où $\mu_A(x)$ est la fonction d'appartenance de x dans A , elle prend des valeurs comprises entre 0 et 1. Dans le cas particulier où μ_A ne prend que des valeurs égales à 0 ou 1, l'ensemble flou A est un ensemble classique de U . Un ensemble classique est donc un cas particulier d'ensemble flou.

L'univers de discours U peut contenir des valeurs continues ou discontinues, donc l'ensemble A de U s'écrit généralement sous la forme :

$$A = \begin{cases} \sum_{x \in U} \mu_A(x) / x & \text{si } U \text{ est discret} \\ \int_U \mu_A(x) / x & \text{si } U \text{ est continu} \end{cases} \quad (4.2)$$

Cette notation est classiquement adoptée pour représenter l'ensemble flou A , bien qu'elle ne fasse référence à aucune idée de sommation ni d'intégration, elle indique pour tout élément x de U son degré $\mu_A(x)$ d'appartenance à A .

4.2.3. Caractéristiques d'un ensemble flou :

Les caractéristiques les plus utilisées pour décrire un ensemble flou sont :

4.2.3.1. Le support :

Le support d'un ensemble flou A , noté $\text{supp}(A)$, est la partie de U sur laquelle la fonction d'appartenance $\mu_A(x)$ n'est pas nulle, il est défini par :

$$\text{supp}(A) = \{ x \in U \mid \mu_A(x) \neq 0 \} \quad (4.3)$$

4.2.3.2. La hauteur :

La hauteur d'un ensemble flou A , notée $h(A)$, est le plus grand degré avec lequel un élément x de U appartient à A , elle est définie par :

$$h(A) = \text{Sup}_{x \in U} (\mu_A(x)) \quad (4.4)$$

Un ensemble flou A de U est dit normal si $h(A) = 1$, autrement il est sous-normale.

4.2.3.3. Le noyau :

Le noyau d'un ensemble flou A , noté $\text{noy}(A)$, est l'ensemble des éléments de U pour lesquels la fonction d'appartenance de A vaut 1 :

$$\text{noy}(A) = \{ x \in U / \mu_A(x) = 1 \} \quad (4.5)$$

4.2.3.4. La convexité :

Un ensemble flou A est dit "convexe" si sa fonction d'appartenance ne contient pas de vallées (Figure 4.1). Formellement, un ensemble A est convexe si et seulement si :

$$\forall u_1, u_2 \in U, \exists \lambda \in [0, 1] : \mu_A(\lambda.u_1 + \lambda.u_2) \geq \min [\mu_A(u_1), \mu_A(u_2)] \quad (4.6)$$

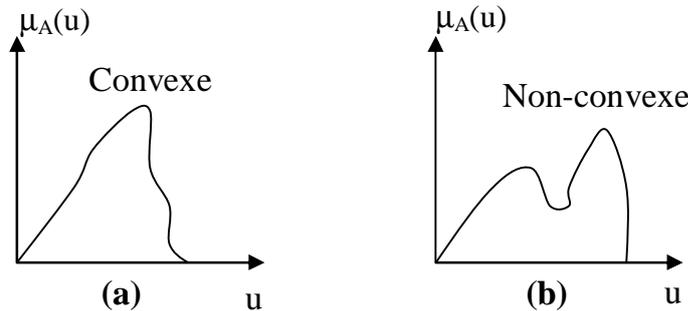


Figure 4.1: Ensemble flou : (a) convexe, (b) Non-convexe

Dans les applications à base de la logique floue, seulement les ensembles flous normaux et convexes sont utilisés.

4.2.4. Fonction d'appartenance d'un ensemble flou (FA) :

Soit A un ensemble flou de U , on attribue à chaque élément $x \in U$, une fonction d'appartenance $\mu_A(x)$ dont la valeur varie entre 0 et 1. La forme des FAs est arbitraire, mais il est raisonnable de prendre des fonctions convexes de façon qu'il existe au moins un point de degré d'appartenance maximal et que le degré décroît quand on s'éloigne de ce point. Les FAs les plus utilisées sont :

- Fonction d'appartenance triangulaire : elle est représentée sur la figure 4.2.
- Fonction d'appartenance trapézoïdale : elle est représentée sur la figure 4.3.
- Fonction d'appartenance gaussienne : (Figure 4.4), elle est donnée par la formule :

$$F(x) = e^{-\left\{ \left(\frac{x-c}{\delta} \right)^2 \right\}} \quad (4.7)$$

Où, c et δ représentent respectivement, le centre et l'écart type de la fonction $F(x)$.

- Fonction d'appartenance en forme de cloche généralisée : (Figure 4.5), elle est donnée par la formule :

$$G(x) = 1 / (1 + \left| \frac{x-c}{a} \right|^{2b}) \quad (4.8)$$

Où c et a représentent respectivement, le centre et la largeur de $G(x)$. Le paramètre b est utilisé pour contrôler la pente au point d'inflexion.

- Fonction d'appartenance sigmoïde : (Figure 4.6), elle est définie par la formule :

$$S(x) = 1 / (1 + e^{-a(x-c)}) \quad (4.9)$$

Où a permet de contrôler la pente au point d'inflexion $x = c$.

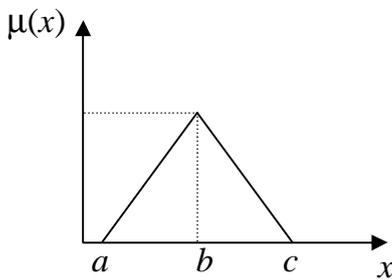


Figure 4.2: Fonction triangulaire

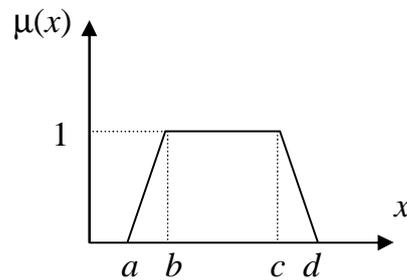


Figure 4.3: Fonction trapézoïdale

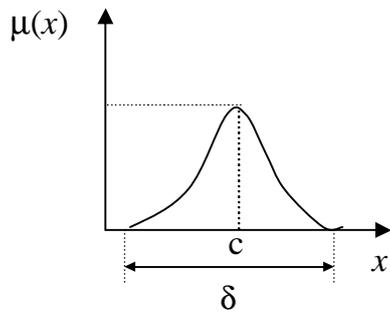


Figure 4.4: Fonction gaussienne.

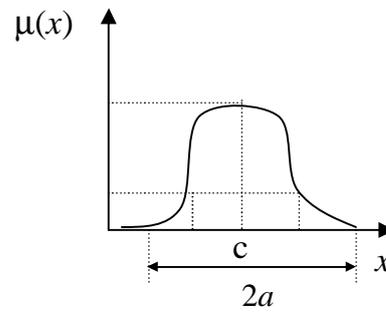


Figure 4.5: Fonction en forme de cloche

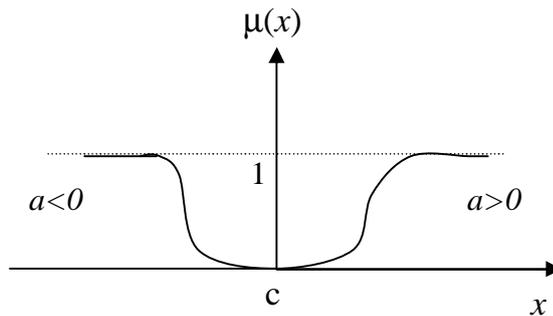


Figure 4.6: Fonction sigmoïde

4.3. Variables floues et linguistiques :

Une variable floue (par exemple, taille) est une variable dont les valeurs sont des termes ou des mots dans le langage naturel (par exemple, petite, moyenne, grande). En général, les valeurs peuvent être des expressions dans un langage spécifié et dans ce cas, la variable est linguistique. Les expressions sont plus utilisées que les nombres puisque les caractéristiques linguistiques sont moins spécifiques que celles numériques.

Exemple : Soit "la vitesse" une variable linguistique, ses valeurs peuvent être "très lente, lente, moyenne, rapide, très rapide".

En général, les valeurs d'une variable linguistique peuvent être générées à partir d'un terme primaire tel que "lente" et des modificateurs "très, peu, assez, moins, pas,..." et les éléments de connexion "et" ou "ou". Une valeur de vitesse peut être "pas très lente et pas très rapide".

4.4. Opérations sur les ensembles flous :

Soient A et B deux ensembles flous définis sur l'univers de discours U , ayant respectivement les fonctions d'appartenance μ_A et μ_B , les opérations de base effectuées sur les ensembles flous sont les suivantes :

4.4.1. L'égalité :

Deux ensembles A et B de U sont égaux, si leurs fonctions d'appartenance prennent la même valeur pour tout élément x de U :

$$\forall x \in U \quad \mu_A(x) = \mu_B(x) \quad (4.10)$$

4.4.2. L'inclusion :

On dit que A est inclus dans B et on note $A \subseteq B$, si leurs fonctions d'appartenance sont telles que :

$$\forall x \in U \quad \mu_A(x) \leq \mu_B(x) \quad (4.11)$$

4.4.3. L'intersection :

L'intersection de deux ensembles flous A et B de U est l'ensemble flou C qui contient tous les éléments x de U appartenant à la fois à A et à B : $C = A \cap B$.

Le degré avec lequel chaque élément x de U appartient à C est le plus petit des degrés avec lesquels il appartient à A et B :

$$\forall x \in U \quad \mu_C(x) = \min(\mu_A(x), \mu_B(x)) \quad (4.12)$$

Où min désignant l'opérateur de minimisation.

4.4.4. L'union :

L'union de deux ensembles flous A et B , noté $A \cup B$, est l'ensemble flou C qui contient tous les éléments x de U appartenant à A ou bien à B . Le degré avec lequel chaque élément appartient à C est le plus grand des degrés avec lesquels il appartient à A ou à B c'est à dire :

$$\forall x \in U \quad \mu_C(x) = \max(\mu_A(x), \mu_B(x)) \quad (4.13)$$

Où max désignant l'opérateur de maximisation.

Comme dans la théorie des ensembles classiques, les définitions que nous venons de donner conduisent aux propriétés suivantes pour tout A et B de U :

- Associativité de \cap et de \cup .
- Commutativité de \cap et de \cup .
- $A \cap U = A, A \cap \emptyset = \emptyset$.
- $A \cup \emptyset = A, A \cup U = U$.
- $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$.
- $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$.

4.4.5. La Complémentation :

Etant donné un ensemble flou A de U , son complément est l'ensemble flou \bar{A} contenant tous les éléments de U n'appartenant pas à A . La fonction d'appartenance de \bar{A} est donnée par :

$$\forall x \in U \quad \mu_{\bar{A}}(x) = 1 - \mu_A(x) \quad (4.14)$$

Puisque la logique floue est une extension de la logique ordinaire, toutes les opérations de la première sont obtenues par analogie à la seconde, sauf pour les lois de contradiction et de tiers exclu qui sont définies comme suit :

- **Loi de contradiction** : Un élément ne peut jamais être en même temps dans l'ensemble et dans son complément :

$$A \cap \bar{A} = \emptyset \quad (4.15)$$

- **Loi du tiers exclu** : L'ensemble et son complément doivent recouvrir l'univers de discours :

$$A \cup \bar{A} = U \quad (4.16)$$

4. 5. Normes et conormes triangulaires :

Les opérations d'intersection, d'union et de complémentation des ensembles flous habituellement employées peuvent être remplacées par d'autres opérations construites à l'aide d'opérateurs différents du minimum, du maximum et de la complémentation à 1. Ces opérateurs ont été introduits dans le domaine des espaces métriques aléatoires, et on fait appel à eux lorsque les opérations habituelles ne s'avèrent pas satisfaisantes.

4.5.1 Normes triangulaires :

Une norme triangulaire T-norme est une fonction définie de $[0,1] \times [0,1] \rightarrow [0,1]$. Elle interprète la conjonction des ensembles flous et inclut :

- Intersection: $x \cap y = \min(x, y)$ (4.17)

- Produit algébrique: $x \bullet y = xy$ (4.18)

- Produit borné: $x \otimes y = \max(0, x + y - 1)$ (4.19)

- Produit drastique : $x \cap \cdot y = \begin{cases} x & y = 1 \\ y & x = 1 \\ 0 & x, y \neq 1 \end{cases}$ (4.20)

4.5.2. Conormes triangulaires :

Une conorme triangulaire T-conorme est une fonction définie de $[0,1] \times [0,1] \rightarrow [0,1]$. Elle interprète la disjonction des ensembles flous et inclut :

- Union: $x \cup y = \max(x, y)$ (4.21)

- Somme algébrique: $x + y = x + y - xy$ (4.22)

- Somme bornée: $x \oplus y = \min(1, x + y)$ (4.23)

- Somme drastique $x \cup \cdot y = \begin{cases} x & y = 0 \\ y & x = 0 \\ 1 & x, y \neq 0 \end{cases}$ (4.24)

Où x et y sont des variables floues. Les T-normes et les T-conormes les plus utilisées sont le min et le max.

4.6. Relations floues :

Parmi les concepts flous les plus importants du point de vue des applications qu'ils peuvent avoir, les relations floues généralisent la notion de relation classiquement définie sur des ensembles. Elles mettent en évidence des liaisons imprécises ou graduelles entre éléments d'un même ensemble [53]. Une relation floue représente donc le degré de présence ou d'absence d'une association, d'interaction ou d'interconnexion entre les éléments de deux ou plusieurs ensembles flous.

Soient U_1, U_2, \dots, U_n , n univers de discours, une relation floue R est un ensemble flou défini par :

$$R = \{ ((u_1, u_2, \dots, u_n), \mu_R(u_1, u_2, \dots, u_n)) \mid (u_1, u_2, \dots, u_n) \in U_1 \times U_2 \times \dots \times U_n \} \quad (4.25)$$

Remarque : Les relations floues étant des cas particuliers d'ensembles flous, toutes les propriétés et définitions qui concernent les ensembles flous leur sont applicables.

4.6.1. Composition de relations floues :

Soient trois ensembles flous A , B et C , la connaissance de deux relations floues l'une entre A et B , l'autre entre B et C permet d'établir une relation entre A et C , comme dans le cas des relations classiques.

Par exemples, si la largeur est très inférieure à la longueur et celle-ci est un peu plus petite que la hauteur, on doit pouvoir mettre en évidence une relation entre la largeur et la hauteur.

La composition de deux relations floues R_1 sur $A \times B$, et R_2 sur $B \times C$ définit une relation $R = R_1 \circ R_2$ sur $A \times C$ de fonction d'appartenance définie par :

$$\forall (a,c) \in A \times C, \mu_R(a,c) = \sup_{b \in B} \min(\mu_{R_1}(a,b), \mu_{R_2}(b,c)) \quad (4.26)$$

Cette définition correspond à la composition max-min, la plus classiquement utilisée, il est cependant possible de remplacer l'opérateur min par un autre opérateur T, par exemple une norme triangulaire, et en particulier le produit pour définir la composition max-T.

4.7. Dédution floue et raisonnement approximatif :

Afin de tirer des conclusions, plusieurs valeurs de variables linguistiques sont liées entre elles par des règles, on parle alors de la déduction floue ou d'inférence. Le raisonnement flou est le processus qui combine des propositions en utilisant les opérations de conjonction, disjonction et d'implication. Une règle floue a la forme suivante :

$$R : \text{SI } (x_1 \text{ est } A_1 \text{ et } x_2 \text{ est } A_2) \text{ ALORS } (y \text{ est } B) \quad (4.27)$$

Où A_1 , A_2 et B sont des variables linguistiques définies sur les univers de discours U_1 , U_2 et U_3 respectivement. Souvent la partie SI(-) est appelée l'antécédente ou la prémise, tandis que la partie ALORS (-) est appelée conclusion ou conséquence, c'est à dire, une règle se compose de paire (prémise, conséquence). Donc le raisonnement approximatif est tout simplement la dérivation des conclusions à partir des règles à une ou plusieurs conditions. Les deux importantes règles de base utilisées dans le raisonnement approximatif sont :

- Les règles conditionnelles d'inférences.
- Le modus Ponens Généralisé (MPG).

La première utilise la relation floue afin de représenter explicitement la connexion entre deux propositions floues. La deuxième utilise la règle " SI –ALORS " qui représente implicitement la relation floue. Pour générer une action bien précise, un mécanisme d'inférence est nécessaire, sa forme la plus simple est la suivante :

$$\begin{array}{l} \text{Prémise 1 : } x_1 \text{ est } A_1 \\ \text{Prémise 2 : } x_2 \text{ est } A_2 \\ \hline \text{Conséquence : } y \text{ est } B \end{array} \left. \vphantom{\begin{array}{l} \text{Prémise 1 : } x_1 \text{ est } A_1 \\ \text{Prémise 2 : } x_2 \text{ est } A_2 \\ \hline \text{Conséquence : } y \text{ est } B \end{array}} \right\} \begin{array}{l} \text{Si } (x_1 \text{ est } A_1 \text{ et } x_2 \text{ est } A_2) \\ \text{Alors } y \text{ est } B \end{array}$$

On peut utiliser différentes prémisses pour obtenir différentes conclusions avec la même implication.

4.8 Commande et Réglage par la Logique Floue

Le principe fondamental de la logique floue est de permettre des informations ambiguës d'être classées dans des ensembles à limites non tranchants, c'est à dire que la forme des fonctions d'appartenance n'est pas rigide et peut dépendre du contexte envisagé, à savoir professionnel, social ou autre. Cette construction permet une description des fonctions de contrôle avec des termes linguistiques vagues, tels que : "Large", "Petit", "Froid", "Rapide", etc. Une des applications de la logique floue est le réglage et la commande des manipulateurs mobiles. Un de ses avantages principaux est de permettre l'incorporation des connaissances d'experts sur le *comment commander le système, sans pour autant connaître la description mathématique du problème*. C'est ainsi que la logique floue se prête particulièrement bien pour des processus mal modélisés.

Nous nous sommes intéressés dans cette thèse à résoudre le problème de commande des systèmes robotiques à l'aide de la logique floue. Cet intérêt est dû principalement au succès extraordinaire qu'a connue la logique floue et qui présente les avantages suivants :

- La non-nécessité d'une modélisation.
- La possibilité d'implémenter les connaissances d'experts.
- La simplicité dans la réalisation
- La robustesse face aux perturbations

On examinera dans cette partie les différentes étapes de la structure des systèmes à base de la logique floue schématisée par la Fig. 2.1.

4.9. Structure des systèmes à base de la logique floue :

Un système à base de la logique floue, ou tout simplement un système flou, est décrit par un ensemble de règles de type SI (conditions) ALORS (conclusion) permettant de convertir la stratégie de contrôle linguistique acquise auprès d'un expert humain en une stratégie de contrôle automatique bien adaptée au monde réel. Il se compose principalement, comme c'est illustré sur la figure 4.7, de quatre blocs essentiels [54] :

- Une interface de fuzzification à l'entrée.
- Une base de connaissances floues.
- Un engin d'inférence ou bloc de prise de décision.
- Une interface de défuzzification à la sortie.

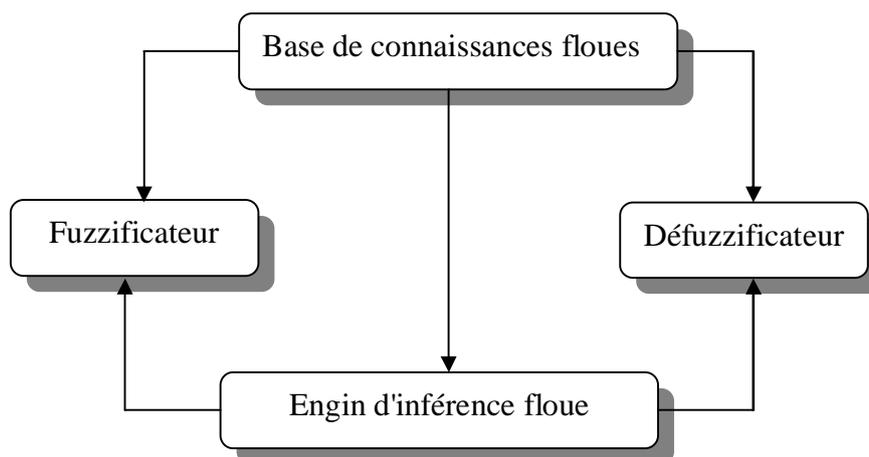


Figure 4.7: Structure de base d'un système flou.

4.9.1. L'interface de fuzzification :

Ce bloc effectue la conversion des grandeurs physiques d'entrées du système (valeurs numériques observées) en variables linguistiques pour permettre leur traitement par l'engin d'inférence.

4.9.2. La base de connaissances floues :

Elle contient les définitions des fonctions d'appartenance des variables d'entrées et de sortie (triangulaire, gaussienne, trapézoïdale, ...etc.), ainsi que les règles floues qui définissent la relation entre l'observation (antécédent) et l'action (conclusion).

4.9.3. L'engin d'inférence :

L'engin d'inférence permet de définir la stratégie de contrôle en utilisant les implications floues qui lient les différentes variables de chaque règle. Les mécanismes d'inférence les plus utilisés sont :

- Mécanisme d'inférence MAX-MIN.
- Mécanisme d'inférence MAX-PROD.
- Mécanisme d'inférence SOM-PROD.

4.9.4. L'interface de défuzzification :

Le bloc de défuzzification constitue le lien de communication entre le monde de raisonnement approximatif (système flou) et le monde réel du processus (système à contrôler). Il fait la conversion des variables linguistiques en valeurs numériques compatibles avec le système à commander. Plusieurs méthodes de défuzzification sont proposées, mais la plus utilisée est celle du centre de gravité qui est définie par :

$$Z = \begin{cases} \frac{\sum_{i=1}^n z_i \cdot \mu(z_i)}{\sum_{i=1}^n \mu(z_i)} & \text{pour un système discret} \\ \frac{\int z \cdot \mu(z) \cdot dz}{\int_w \mu(z) \cdot dz} & \text{pour un système continu} \end{cases} \quad (4.28)$$

Où n est le nombre de niveaux de quantification de la sortie.

4.10. Conclusion :

Les applications de la logique floue touchent tous les domaines parce qu'elle s'efforce d'apporter des solutions à un problème clé de toutes les réalisations pratiques, d'une part l'appréhension du monde réel est imparfaite en raison de sa grande complexité, d'autre part sa représentation doit à la fois conserver suffisamment d'information pour ne pas trahir les données réelles et la simplifier pour être manipulable efficacement. Dans certains cas, elle permet de résoudre des problèmes simples pour une mise en place rapide de techniques éprouvées. Dans d'autres cas, une solution spécifique doit être élaborée pour traiter un problème complexe. C'est ce qui fait la diversité des applications industrielles mises en évidence, qui vont des appareils électroménagers les plus variés au contrôle des usines chimiques ou au guidage de véhicules sans pilote, par exemple.

D'une façon plus générale, les ensembles flous peuvent intervenir efficacement dans la modélisation d'un système complexe, principalement en raison de leur capacité à synthétiser des informations, à permettre une approche globale de certaines caractéristiques du système grâce à la gradualité qui leur est inhérente et également, bien sûr, en raison de leur aptitude à traiter des connaissances imparfaites, c'est à dire incomplètes, approximatives, vagues ou soumises à des erreurs de mesure.

Chapitre 5

Planification de mouvement des robots manipulateurs.

5.1. Introduction :

Sur un poste de travail, un robot est amené à réaliser trois grandes classes de tâches : des déplacements purs, des efforts purs et des tâches compliantes combinant des déplacements et des efforts. Dans ce chapitre, nous abordons le problème de la génération (ou planification) de mouvement du robot dans le cas seulement de déplacements purs.

Lors d'une tâche de déplacement, on attend du robot qu'il suive, avec une loi horaire donnée, une trajectoire définie par une suite de repères abusivement appelés points, correspondant aux situations successives de l'organe terminal du robot, ces points peuvent être [44] :

- Programmés par apprentissage,
- Donnés en ligne par un capteur extéroceptif,
- Générés par un planificateur de mouvement.

Le problème de génération de mouvement est de calculer la consigne de la commande qui assure le passage du robot par ces points, on peut envisager l'une des solutions suivantes :

- Considérer comme consigne les coordonnées articulaires correspondant aux situations désirées, l'asservissement assure l'arrivée de chaque articulation à sa valeur finale.
- Trouver une ou plusieurs fonctions mathématiques d'interpolation qui assurent le passage du robot par ces points de la trajectoire en fonction du temps.

5.2. Problématique de génération de mouvement :

Avant tout mouvement et action, la stratégie à suivre par le robot doit être planifiée, cette stratégie peut être totalement spécifiée, partiellement spécifiée ou non spécifiée au système de commande [47].

5.2.1. Trajectoires totalement spécifiées :

Ici, l'opérateur réalise une programmation par apprentissage en faisant un enregistrement continu de la trajectoire, cette dernière est totalement spécifiée au calculateur, toutes les contraintes sont prises en compte durant la phase d'apprentissage. Durant la phase d'exécution, le calculateur restitue la trajectoire telle qu'elle a été enregistrée sans avoir connaissance à des contraintes existantes (Figure 5.1).

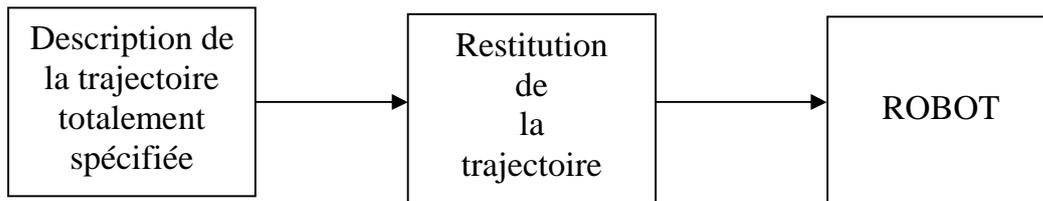


Figure 5.1: Génération de la trajectoire totalement spécifiée.

5.2.2. Trajectoires partiellement spécifiées :

Lorsque le calculateur dispose de configurations importantes que le robot doit atteindre, il doit générer une trajectoire qui permet de passer par ces configurations. Pour aller d'une configuration importante à une autre, le nombre de trajectoires possibles est infini, cependant des contraintes d'ordres divers permettent de choisir ou imposent une trajectoire à suivre.

5.2.3. Trajectoires non spécifiées :

Dans le cas où la trajectoire n'est pas spécifiée, le calculateur doit posséder un générateur de plan lui permettant de générer les configurations importantes et les trajectoires qui les lient, à partir de la connaissance de l'environnement et des données de la description de la tâche.

La figure 5.2 présente les principales étapes nécessaires à la génération d'une trajectoire non spécifiée.

C'est cette dernière solution qui est développée au cours de ce chapitre, mais avant d'en présenter les méthodes, nous analysons les raisons qui peuvent présider au choix d'une génération de mouvement dans l'espace des configurations ou dans l'espace opérationnel.

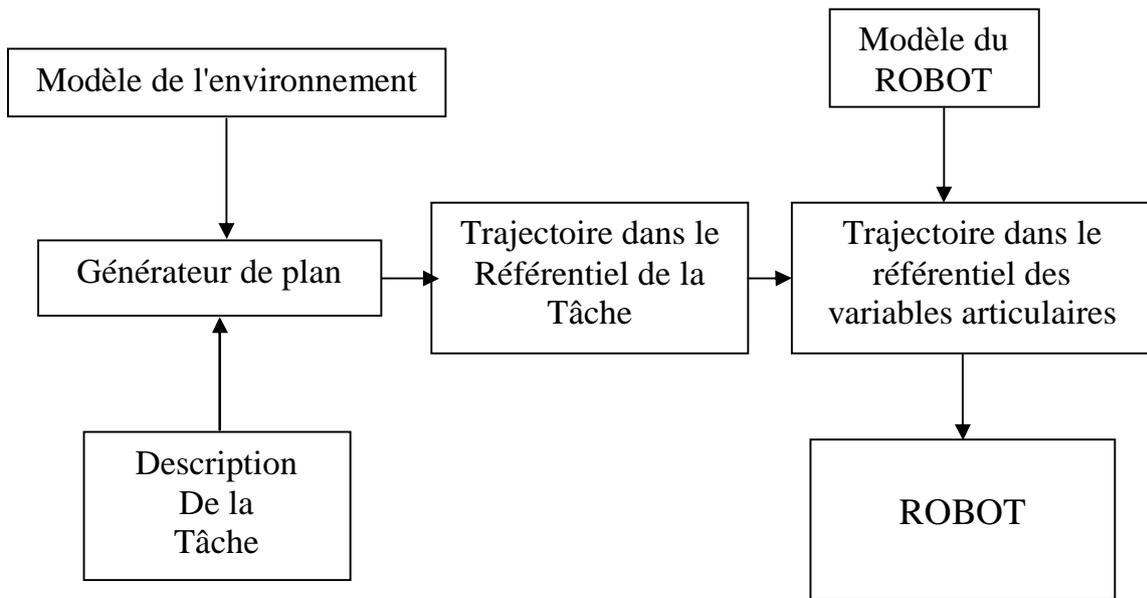


Figure 5.2: Génération de la trajectoire non-spécifiée

Parmi les tâches, de déplacement pur que le robot peut réaliser, on peut citer :

- Les mouvements entre deux points, la trajectoire n'étant pas contrainte entre les points (mouvement libre).
- Les mouvements entre deux points via des points intermédiaires, spécifiés notamment pour éviter les obstacles, la trajectoire est libre entre les points intermédiaires.
- Les mouvements entre deux points, la trajectoire étant contrainte entre les points (trajectoire rectiligne, par exemple).
- La poursuite d'un objet mobile sur un convoyeur ou, plus généralement, l'asservissement des mouvements du robot à des consignes issues de capteurs extéroceptifs ...etc.

Dans les deux premiers cas, la génération de mouvement peut se faire directement dans l'espace des configurations; elle se traduit par une séquence de positions articulaires (voire de vitesse et d'accélération) constituant les consignes des asservissements. Dans les trois derniers cas, la trajectoire étant décrite dans l'espace opérationnel, il est préférable de raisonner dans cet espace. La loi de commande engendrée doit ensuite être transformée en consignes articulaires par le changeur de coordonnées.

Ces deux approches de génération de mouvement dans l'espace articulaire et dans l'espace opérationnel sont schématisées sur les figures (5.3 et 5.4).

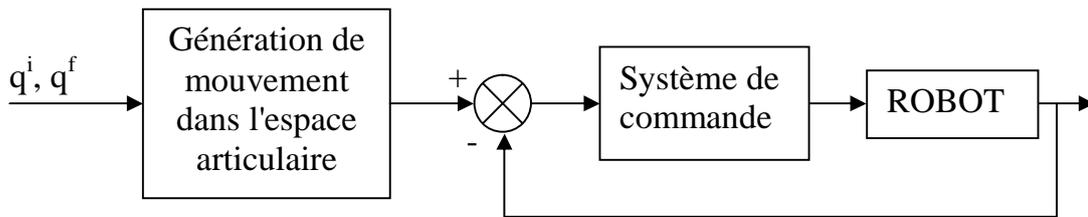


Figure 5.3: Génération de mouvement dans l'espace articulaire

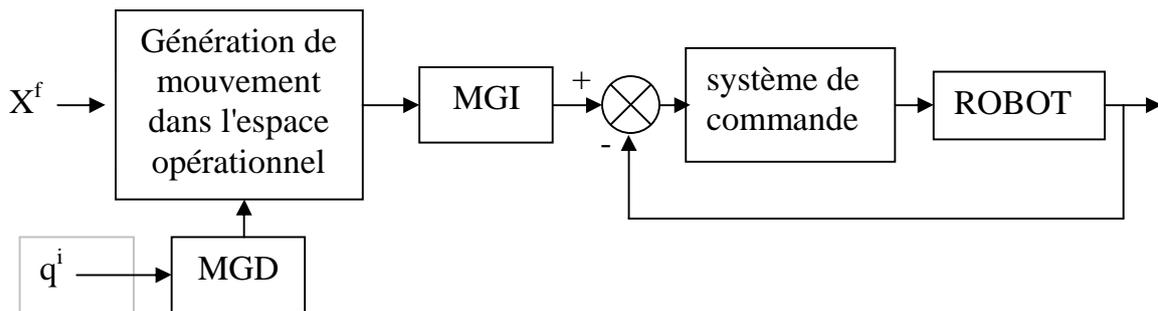


Figure 5.4: Génération de mouvement dans l'espace opérationnel

Leurs principales caractéristiques sont décrites dans ce qui suit :

5.2.4. Génération de mouvement dans l'espace articulaire :

Une séquence de variations des positions articulaires est générée directement dans l'espace articulaire, elle présente plusieurs avantages :

- Le mouvement est minimal sur chaque articulation.
- Elle nécessite moins de calcul en ligne, puisqu'il n'y a pas d'appel au changeur de coordonnées.
- Le mouvement n'est pas affecté par le passage sur les configurations singulières.
- Les contraintes de vitesses et de couples maximaux sont connues avec précision puisqu'elles correspondent aux limites physiques des actionneurs.

En contrepartie, la géométrie de la trajectoire dans l'espace opérationnel ne peut pas être imposée. Entre deux points donnés, l'organe terminal se déplace de façon imprévisible mais répétitive; il y a donc risque de collision lorsque le robot évolue dans un environnement encombré. Ce type de mouvement est par conséquent approprié pour réaliser des déplacements rapides dans un espace dégagé.

5.2.5. Génération de mouvement dans l'espace opérationnel :

La génération de mouvement dans l'espace opérationnel permet de contrôler la géométrie de la trajectoire par contre :

- Elle implique la transformation en coordonnées articulaires de chaque point de la trajectoire,
- Elle peut être mise en échec lorsque la trajectoire calculée passe par une position singulière,
- Elle est mise en échec chaque fois que les points de la trajectoire engendrés ne sont pas dans le volume accessible du robot,
- Les limites en vitesse et en couple des articulations étant définies dans l'espace articulaire, ces contraintes ne sont pas directement utilisables dans l'espace de la tâche, on exprime alors ces limites par des valeurs traduisant les performances moyennes satisfaites quelle que soit la configuration du robot, on impose donc à ce dernier de travailler au-dessous de ses moyens réels.

Le choix d'une méthode de génération de mouvement dépend de l'application considérée. Chaque approche a ses propres limites inhérentes au fait que les contraintes sont exprimées soit dans l'espace articulaire (vitesses, couples, butées, ...), soit dans l'espace opérationnel (précision, obstacles, ...).

5.3. Les approches de planification de mouvement :

Plusieurs techniques de planification de mouvement ont été développées et classées en deux grandes catégories :

5.3.1. Les techniques classiques :

La plupart de ces techniques utilisent des fonctions mathématiques d'interpolations pour lier les deux positions initiale et finale, ceci en minimisant le temps du déplacement. On cite, à titre d'exemple, quelques techniques qui ont été appliquées avec succès : la génération de mouvement par la loi optimale [55-57], planification de trajectoires en utilisant le critère du temps minimal [58-59] et d'autres méthodes de génération de mouvement sont présentées dans [60-68]. Ces méthodes classiques permettent de réaliser des déplacements rapides, mais elles présentent l'inconvénient qu'elles ne sont utilisables que dans les environnements dégagés.

5.3.2. Techniques récentes :

Dès l'apparition des réseaux de neurones et des algorithmes génétiques, le monde d'optimisation et d'identification a connu un changement radical, ceci est dû à leur capacité importante de traitement des problèmes complexes.

L'utilisation des réseaux de neurones pour la conception des planificateurs de mouvements est très importante notamment, dans les problèmes de prédiction de trajectoires, par exemple dans le cas de poursuite d'objets mobiles, où on a besoin de prédire la trajectoire en temps réel pour réaliser la saisie de ces objets sans échec. Plusieurs études sont développées dans cette optique : [68-74].

Les algorithmes génétiques sont bien connus des procédures d'optimisation très puissantes, leur utilisation dans la résolution des problèmes de planification de mouvement des robots manipulateurs est très intéressante [74-76], mais elle nécessite la transformation du problème de planification de mouvement en un problème d'optimisation d'une fonction d'adaptation bien définie. Cette fonction doit prendre en compte certaines contraintes pour que le robot puisse exécuter le déplacement en évitant ou en contournant les obstacles qui peuvent être les objets à manipuler ou d'autres éléments présents dans l'environnement mais sur lesquels le robot n'agit pas.

5.4. Conception d'un planificateur de mouvement par les AGs :

La tâche principale d'un planificateur de mouvement est de générer la trajectoire satisfaisant les deux contraintes suivantes [69] :

- 1- La trajectoire générée par le planificateur de mouvement doit être optimale présentant le plus court déplacement sur les articulations du robot et par conséquent un temps de déplacement minimal.
- 2- La trajectoire générée par le planificateur de mouvement doit permettre au robot de réaliser le déplacement désiré sans collision avec d'éventuels obstacles existant dans l'espace de travail du robot.

Pour satisfaire ces deux contraintes, nous devons choisir la méthode adéquate parmi celles précédemment citées, le choix d'une méthode ou d'une autre dépend étroitement de la nature du problème à traiter, du type de données disponibles et des performances désirées.

Dans ce mémoire, nous avons voulu apporter notre contribution à l'étude et l'analyse du problème de planification de mouvement des robots manipulateurs en utilisant la notion

des algorithmes génétiques. Un système de planification de mouvement peut être schématisé comme suit : (Figure 5.5)

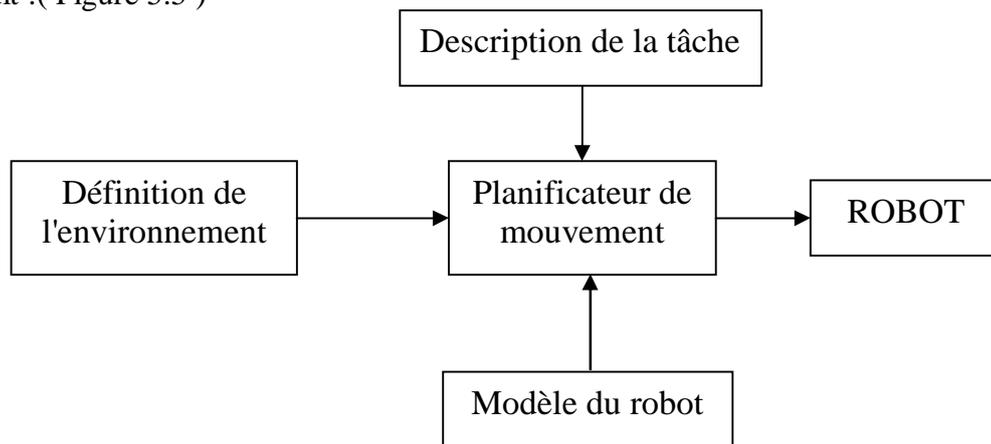


Figure 5.5: Structure du planificateur de mouvement

5.4.1. Formulation du problème de planification de mouvement :

La planification de mouvement des robots manipulateurs peut être divisée en deux sous problèmes : le premier est la recherche de la trajectoire optimale permettant la réalisation du mouvement dans un temps minimal. Le deuxième est l'optimisation de la séquence de variations des angles des articulations permettant au robot de se mouvoir sans collision avec les obstacles existants dans son environnement de travail.

La position du joint i du robot est donnée par la relation suivante (Figure 5.6) :

$$z_i = z_{i-1} + r_i \cdot e^{j \cdot \theta_i} \quad (5.1)$$

Avec $z_i = x_i + j \cdot y_i$ et $j^2 = -1$ (5.2)

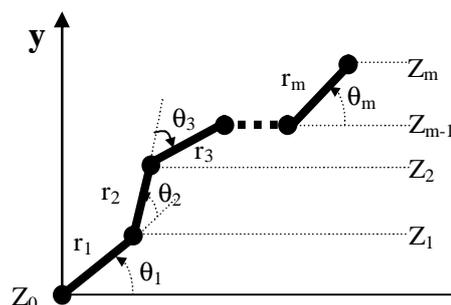


Figure 5.6: Les positions des joints dans l'espace cartésien

Où z_{i-1} est la position du joint $(i-1)$, r_i est la longueur du segment i et θ_i est l'angle du joint i .

L'espace occupé par le segment (i) entre z_{i-1} et z_i est donné par la relation suivante :

$$D_i = \{ t.z_i + (1-t).z_{i-1} ; 0 \leq t \leq 1 \} \quad (5.3)$$

Avec : t est un paramètre.

L'espace total occupé par le robot est donné par :

$$D_t = D_1 \cap D_2 \cap \dots \cap D_m = \bigcap_{i=1}^m D_i \quad (5.4)$$

Où m est le nombre de joints du robot manipulateur.

Le robot peut se déplacer sans collision si :

$$D_t \subset D_f \quad (5.5)$$

Où D_f est l'espace vacant d'obstacles.

On note $\theta = (\theta_1, \theta_2, \dots, \theta_m)^T$, si θ^* qui satisfait la relation (5.5) existe, le mouvement sans collision est possible.

Pour générer θ^* , nous avons utilisé un AG décrit comme suit :

5.4.2. Description de l'AG proposé :

Les algorithmes génétiques sont des procédures d'optimisation très puissantes, ils risquent moins d'être piégés dans des minima locaux à cause de l'exploration parallèle d'un ensemble de solutions pour un problème donné [48].

Dans ce mémoire, nous avons utilisé un AG pour optimiser la séquence de variations des angles des joints permettant au robot de se mouvoir dans son espace de travail sans collision avec les obstacles et en suivant des trajectoires optimales.

Les opérations génétiques utilisées dans notre méthode sont expliquées comme suit :

5.4.2.1. Le codage :

La structure du chromosome sélectionnée est de deux dimensions (Figure 5.7), où les lignes représentent les angles des joints et les colonnes représentent les instants d'échantillonnage. La première colonne représente les positions articulaires initiales des joints du robot, tandis que la $n^{\text{ème}}$ colonne représente les positions articulaires désirées (ou finales). Chaque chromosome représente une trajectoire complète permettant au robot de se déplacer de la configuration initiale vers la destination finale. La longueur du chromosome est variable, ce qui permet d'avoir plusieurs variantes de trajectoires possédant différents temps de déplacement.

		Instants d'échantillonnage			
		1	2	3	n
Joints	1	-13	-11	-8	2
	2	6	23	-13	19
	3	-25	17	-8	0
	m	11	0	-7	-29

Figure 5.7: Structure du chromosome.

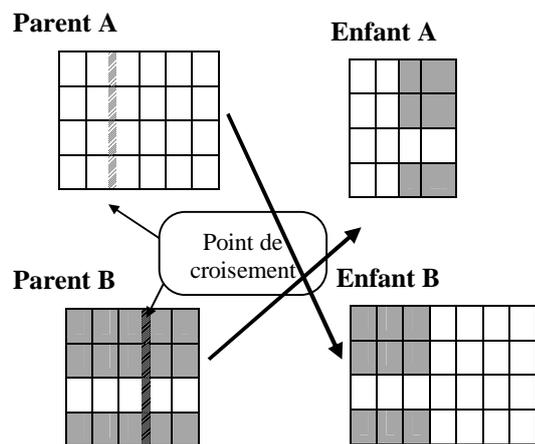
5.4.2.2. La sélection :

L'algorithme génétique proposé utilise la sélection par roue de loterie avec remplacement, cette opération permet aux individus de bonnes qualités de participer à la génération de la prochaine population le plus souvent que ceux de qualités moindres.

5.4.2.3. Le crossover :

L'opérateur de crossover utilisé dans notre méthode est appelé crossover analogique car le point de crossover est choisi par la fonction phénotype est non pas par la fonction génotype (Figure 5.8).

Génotype:



Phénotype:

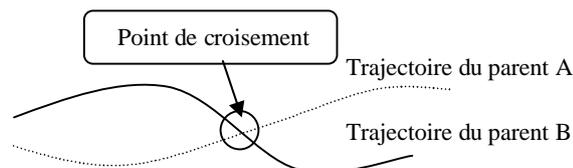


Figure 5.8: Crossover analogique.

La procédure de crossover utilisée dans notre méthode est la suivante :

1. Sélectionner aléatoirement deux parents (parent A et parent B).
2. Choisir aléatoirement un point de crossover pour le parent A.
3. Calculer la position de l'organe terminal du robot au point de crossover choisi du parent A.
4. Comparer la position de l'organe terminal à tous les instants d'échantillonnage du parent B avec la position calculée dans (3).
5. Sélectionner un point de crossover pour le parent B, qui est le plus proche du point de crossover du point A,
6. Réaliser l'échange partiel des gènes entre les deux parents.

5.4.2.4. La mutation addition :

Cette opération insère une longueur aléatoire de gènes à partir d'un point choisi aléatoirement (Figure 5.9), par cette opération, de nouvelles trajectoires sont ajoutées au phénotype de l'individu.

5.4.2.5. La mutation soustraction:

Cette opération soustrait à partir d'un point aléatoire, une longueur aléatoire de gènes (Figure 5.10). Cette opération permet de réduire la longueur du chromosome et donc la longueur de la trajectoire.

Génotype:

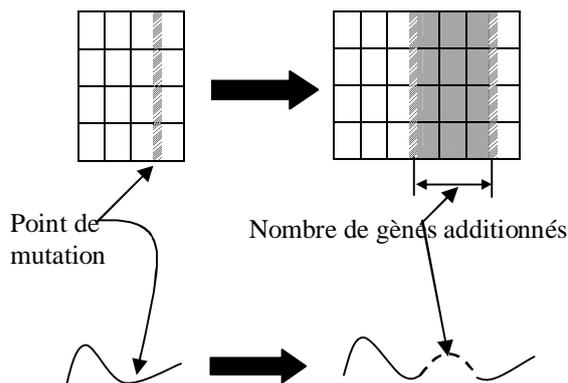


Figure 5.9: Mutation addition

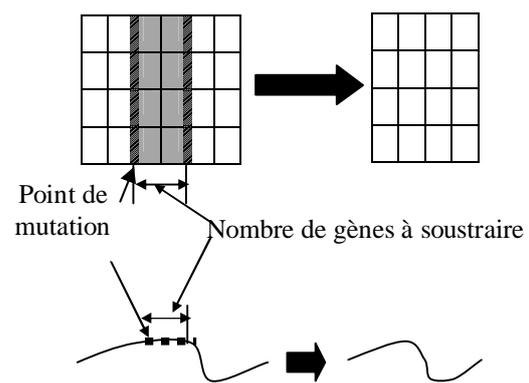


Figure 5.10: Mutation soustraction

5.4.2.6. Introduction des contraintes :

Les algorithmes génétiques sont par nature des techniques d'exploration non-contraintes, cependant, la plupart des problèmes réels contiennent une ou plusieurs contraintes qui doivent être satisfaites. C'est pourquoi on fait appel souvent aux méthodes de pénalisation qui transforment le problème d'optimisation contraint en un problème non-contraint en associant un coût ou une pénalité à chaque violation de contrainte. Ce coût est inclus dans l'évaluation de la fonction d'objectif pour refléter à la fois la qualité et l'acceptabilité des solutions [48].

Dans notre cas la contrainte qui doit être satisfaite est la relation (5.5) du paragraphe (5.4.1).

5.4.2.7. La fonction d'objectif :

L'AG a pour but de générer une solution optimale satisfaisant la minimisation de la fonction donnée par :

$$J = c_1 \cdot L_c + c_2 \cdot \sum_{i=1}^m \sum_{j=2}^n |\theta_i(j) - \theta_i(j-1)| + c_3 \cdot \text{cnt} \quad (5.6)$$

Le terme $\sum_{i=1}^m \sum_{j=2}^n |\theta_i(j) - \theta_i(j-1)|$, représente la somme des variations des angles des

jointes, sa minimisation permet d'avoir une trajectoire lisse. L_c représente la longueur du chromosome, sa minimisation permet d'avoir une trajectoire courte, donc un temps de déplacement minimal, les coefficients de pondération c_1 , c_2 et c_3 sont choisis empiriquement pour éviter les débordements.

Le terme **cnt**, représente le code de pénalisation, sa valeur est donnée par :

$$\text{cnt} = \begin{cases} 0 & \text{Si aucune contrainte n'est violée} \\ n & \text{Si au moins une contrainte est violée} \end{cases} \quad (5.7)$$

Où n est le nombre de contraintes violées.

Les algorithmes génétiques sont par nature des procédures de recherche de maximum, donc il faut transformer le problème de minimisation de J en un problème de maximisation de la fonction f donnée par la relation suivante :

$$f = \frac{1}{1+J} \quad (5.8)$$

L'organigramme suivant représente le principe de fonctionnement de notre système de planification de mouvements (Figure 5.11).

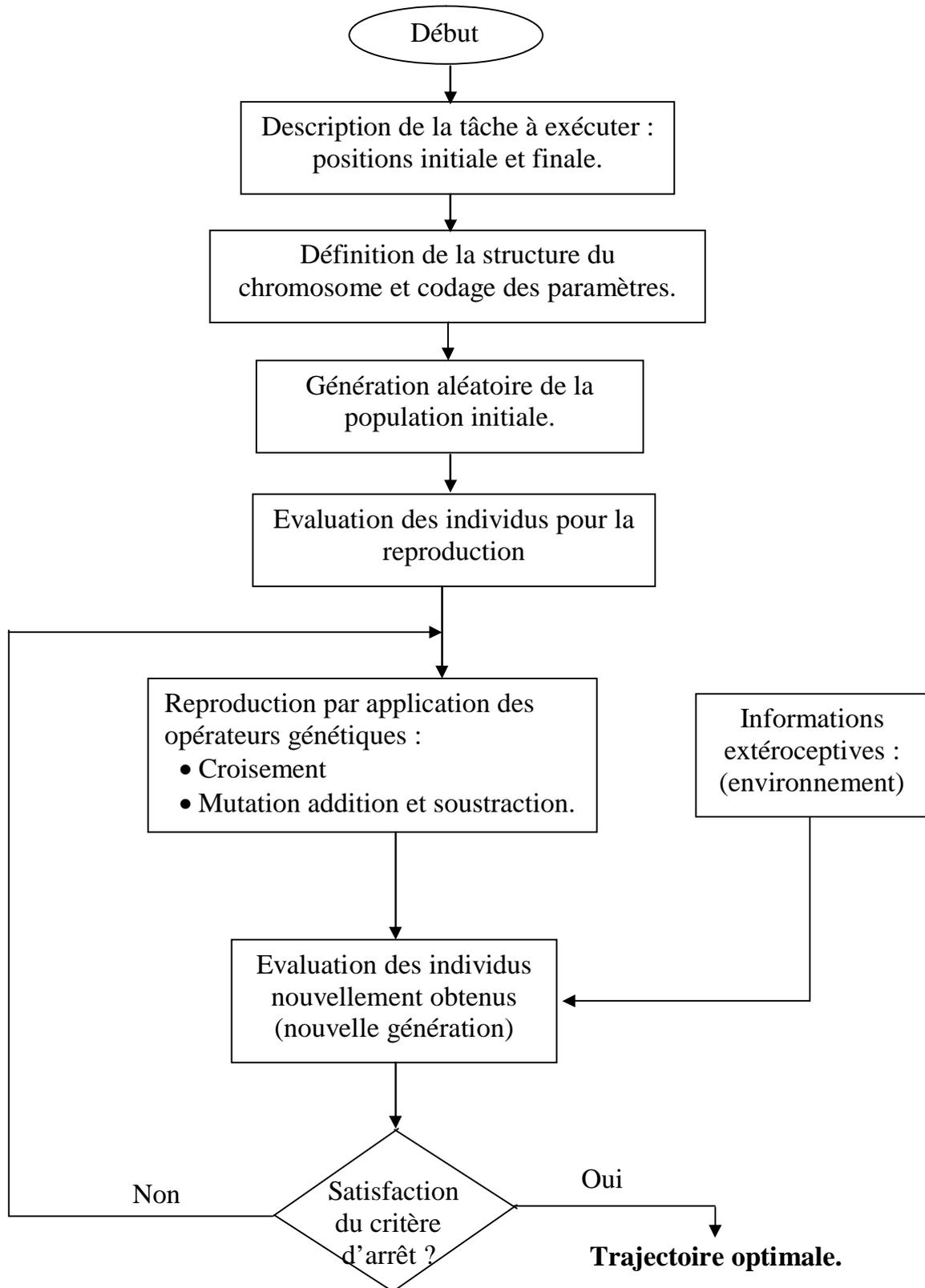


Figure 5.11: Organigramme du planificateur de mouvement.

5.5. Résultats de simulation :

Dans cette section, on va discuter les résultats de simulation du planificateur de mouvement proposé sur un robot manipulateur de type PLANAR à deux degrés de liberté dans un environnement de travail encombré.

La structure géométrique du robot choisi est illustrée sur la figure 5.12, les longueurs des segments du robot sont : $L_1 = 1m$ et $L_2 = 1m$.

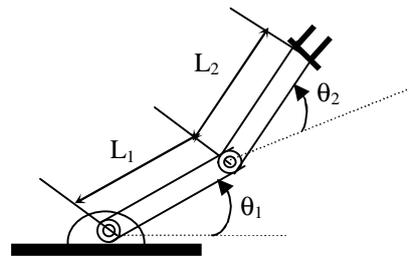


Figure 5.12: Structure du robot à 2 d.d.l.

Puisque le robot possède deux degrés de liberté, la structure du chromosome pour cette application est donnée par :

		Instants d'échantillonnage			
		1	2	3	n
Joints	θ_1	θ_{1i}	14	20	θ_{1f}
	θ_2	θ_{2i}	-27	-12	θ_{2f}

Figure 5.13: Structure du chromosome généré par l'AG

Le nombre de lignes est fixe, égal à 2, tandis que le nombre de colonnes est variable et dépend de la longueur du déplacement (distance entre les configurations initiale et finale) d'un coté, et de l'existence ou non des obstacles dans l'espace de travail du robot de l'autre coté.

La position articulaire initiale du robot manipulateur est $(\theta_1, \theta_2) = (30^\circ, 30^\circ)$, où θ_1 et θ_2 sont les angles des joints. La position initiale de l'organe terminal du robot dans l'espace cartésien est (1.366, 1.366), la position désirée est (1.366, -1.366) dans l'espace cartésien et $(-60^\circ, 30^\circ)$ dans l'espace articulaire [74].

Les paramètres de l'algorithme génétique utilisé sont :

- La taille de la population est fixée à 100.

- Le nombre maximal de générations est égal à 200.
- Les probabilités de la mutation addition et soustraction sont égales à 0.01 et 0.6.
- La probabilité de croisement est égale à 0.9.
- Les constantes de pondération c_1 , c_2 et c_3 valent : 10^{-2} , 10^{-3} et 10^{-1} respectivement.

Le problème posé est de générer la trajectoire optimale permettant de déplacer l'organe terminal du robot de la position initiale (1.366, 1.366) à la position finale (1.366, -1.366) dans un temps minimal et sans collision avec les obstacles.

Dans un premier temps, aucun obstacle n'est introduit dans l'espace de travail du robot, l'évolution de la meilleure valeur de la fonction d'objectif dans la population en fonction du nombre de générations, représentée sur la figure (5.14), est le meilleur résultat en terme de valeur finale de la fonction d'objectif obtenu après 5 exécutions de l'algorithme génétique.

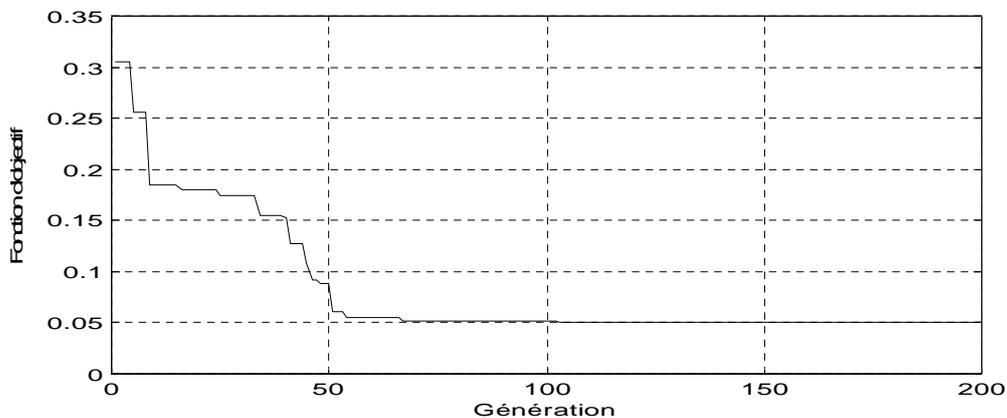


Figure 5.14: Evolution de la fonction d'objectif

Les figures (5.15-16) représentent la trajectoire optimale générée par l'algorithme génétique dans l'espace cartésien et dans celui des articulations. On constate que l'AG proposé arrive à déplacer l'organe terminal du robot de la position initiale à la position désirée en suivant la trajectoire la plus courte.

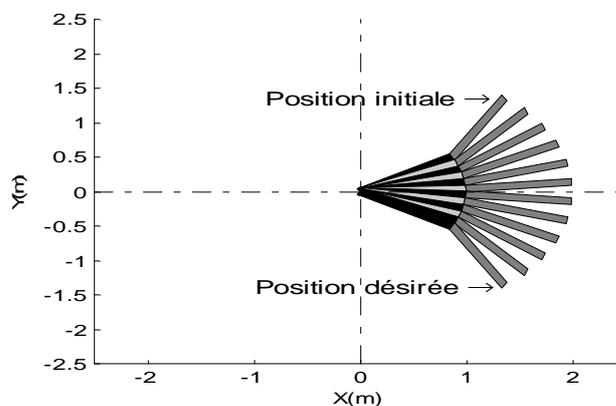


Figure 5.15: Trajectoire optimale non contrainte

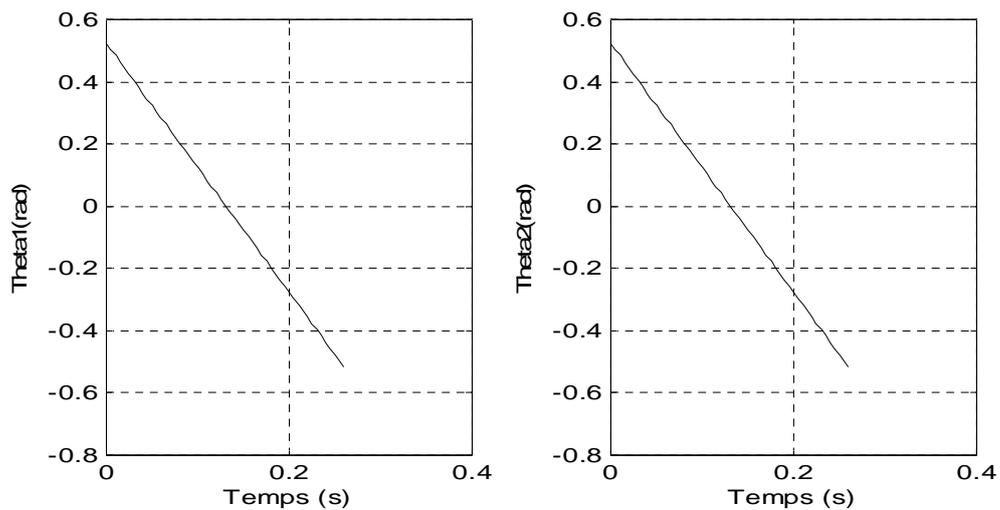


Figure 5.16: Séquence de variations des angles des joints.

Par la suite, un obstacle dont la position est $(0.8, 0)$ est introduit dans l'espace de travail du robot, Les figures (5.17-18) représentent la trajectoire optimale générée par l'algorithme génétique, il est clair que cette trajectoire a permis au robot de déplacer son organe terminal de la position initiale vers celle désirée en évitant la collision avec l'obstacle.

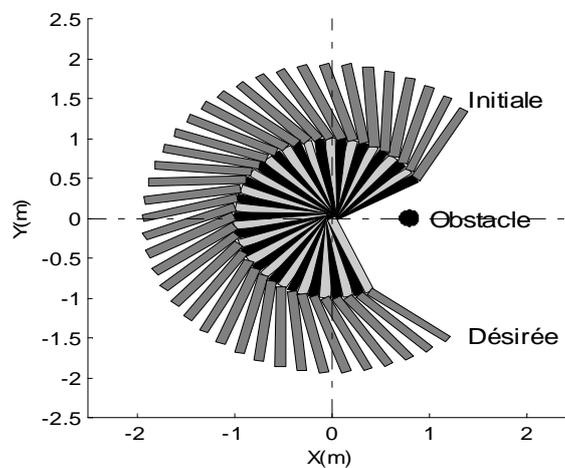


Figure 5.17: Trajectoire optimale contrainte avec un obstacle

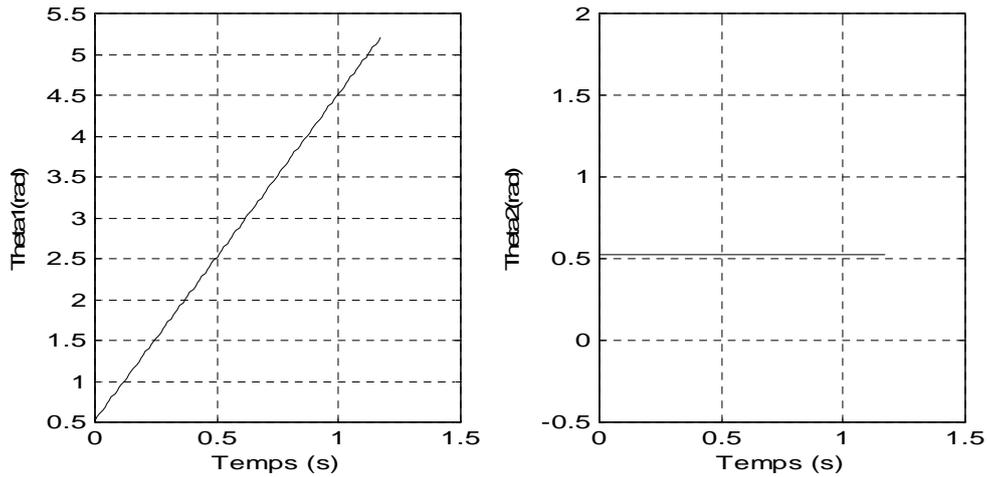


Figure 5.18: Séquence de variations des angles des joints avec un obstacle.

Ensuite, un deuxième obstacle est introduit dans l'espace de travail du robot dont la position est (-1.5,0), les figures (5.19-20) représentent la trajectoire optimale générée par l'algorithme génétique, il apparaît bien que le robot ait réussi de déplacer son organe terminal de la position initiale à la position désirée sans collision avec les obstacles.

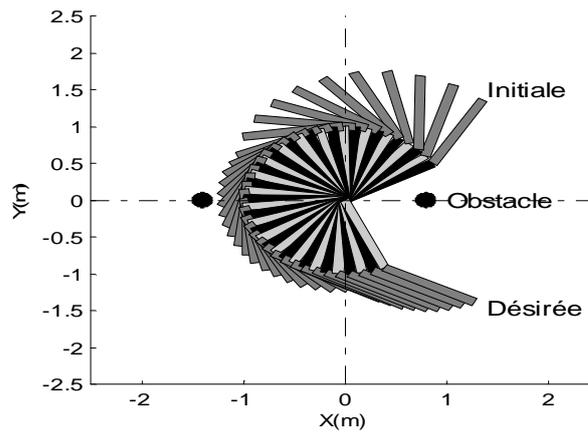


Figure 5.19: Trajectoire optimale contrainte avec deux obstacles.

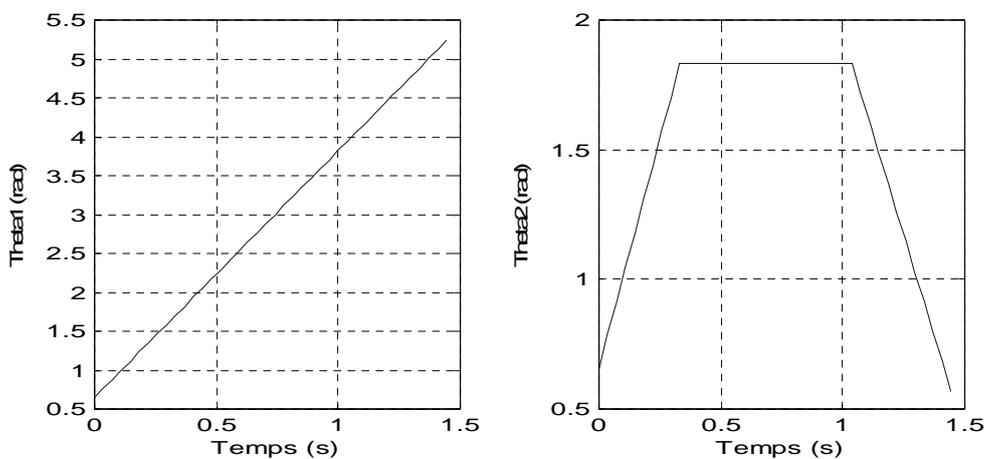


Figure 5.20: Séquence de variations des angles des joints avec deux obstacles.

Finalement, la position du deuxième obstacle est changée par $(-0.8, 0)$, Les figures (5.21-22) représentent la trajectoire optimale générée par l'algorithme génétique. Dans ce cas, le robot n'arrive jamais à déplacer son organe terminal de la position initiale vers celle désirée, parce que les deux obstacles l'empêchent.

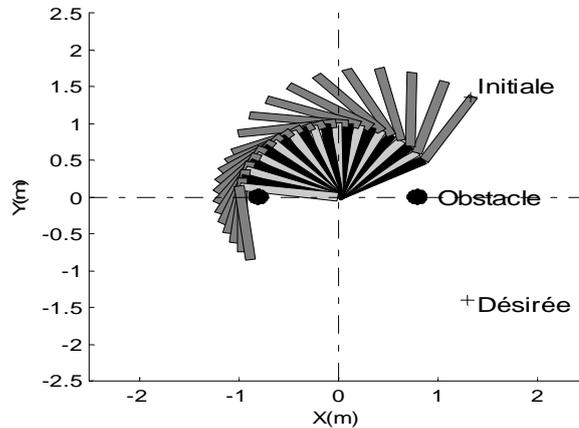


Figure 5.21: Trajectoire optimale contrainte avec changement d'un des obstacles.

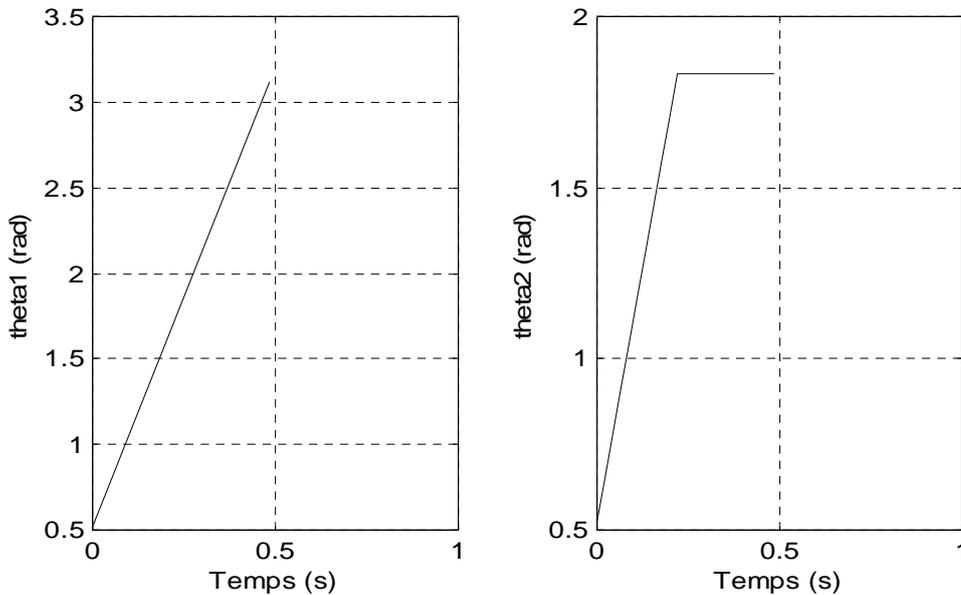


Figure 5.22: Séquence de variations des angles des joints après changement des obstacles.

Ces exemples représentent les cas les plus critiques des obstacles dont les positions sont choisies d'une manière très soignée pour évaluer réellement les performances d'optimisation du planificateur de mouvement proposé.

Les résultats obtenus, montrent clairement que la méthode proposée est très efficace même dans des environnements très encombrés.

5.6. Conclusion :

Dans cette section, nous avons étudié le problème de planification de mouvement qui consiste à générer la trajectoire optimale permettant de déplacer l'organe terminal du robot de la position initiale à la position désirée dans un temps minimal et sans collision avec les obstacles, ceci en utilisant les algorithmes génétiques comme procédures d'optimisation, les opérations génétiques utilisées dans cette étude ne sont pas standards, elles sont très efficaces, notamment, la mutation soustraction qui permet de réduire considérablement la longueur du chromosome et par conséquent, la longueur de la trajectoire à poursuivre se trouve optimisée. La méthode proposée peut être appliquée dans différents espaces de travail possédant différents arrangements d'obstacles.

Chapitre 6

Contrôle de mouvement des robots manipulateurs

6.1. Introduction :

Contrôler un système complexe, fortement non-linéaire ou difficile à modéliser présente une tâche très délicate, ainsi les performances désirées deviennent de plus en plus sévères, c'est pour ces raisons qu'apparaissent de nouvelles méthodes de contrôle plus sophistiquées. L'apparition du formalisme de la logique floue a donné naissance à une approche très efficace où les lois de commande conventionnelles sont remplacées par une série de règles linguistiques déterminant la stratégie et l'objectif de contrôle. Cette approche exige la disponibilité de l'expertise humaine ce qui influence négativement sur les performances des systèmes flous conçus. Ce problème a conduit au développement des systèmes hybrides permettant de réunir les capacités d'optimisation des algorithmes génétiques et celles de décision des systèmes flous. L'utilisation conjointe de ces approches permet de réaliser un système de contrôle flou très efficace et robuste bien adapté à la complexité des systèmes réels (Robot manipulateur).

6.2. Les approches de conception des contrôleurs de mouvement :

Plusieurs méthodes de contrôle de mouvement basées sur diverses approches ont été développées, dans [77-88] des études sont faites pour concevoir des contrôleurs adaptatifs qui peuvent compenser partiellement les incertitudes de la dynamique du robot, mais ce-ci nécessite l'estimation en temps réel de certains paramètres qui est souvent très compliquée et exige un temps de calcul très important. L'association des réseaux de neurones avec les régulateurs classiques PD et PID donne de bons résultats, mais reste incapable de compenser les changements rapides des paramètres dynamiques du robot [89-96].

Bien que les contrôleurs à logique floue ont été appliqués avec succès sur plusieurs procédés industriels complexes, leur conception reste cependant une tâche très difficile. L'approche traditionnelle pour la conception floue basée sur les connaissances acquises par

des opérateurs experts est laborieuse et consomme beaucoup de temps, en plus, cette approche présente d'autres inconvénients tels que :

- Les opérateurs ne peuvent pas facilement transformer leurs connaissances et expériences en une forme algorithmique ou base de règles nécessaires pour la conversion en une stratégie de contrôle automatique.
- Le domaine d'expertise n'est pas toujours disponible.

Suite à ces inconvénients, des recherches approfondies ont été effectuées dans le but d'élaborer des méthodes systématiques et optimales pour la conception des contrôleurs flous, ces recherches ont conduit au développement de deux nouvelles approches :

6.2.1. Approche connexioniste :

Elle consiste à combiner les réseaux de neurones artificiels et les systèmes flous pour construire ce qu'on appelle les réseaux de neurones flous [97-102].

6.2.2. Approche directe :

Elle consiste à appliquer un algorithme d'optimisation pour la conception d'un système flou. A cause de la complexité de l'espace de recherche, les algorithmes génétiques sont utilisés dans la plupart des cas [103-108].

C'est cette dernière approche qui a été utilisée dans ce mémoire pour la conception d'un système de contrôle flou robuste et efficace.

6.3. Conception d'un contrôleur flou par les AGs :

Le problème de conception d'un contrôleur flou (FLC) revient à la détermination des espaces de discours des variables d'entrées et de sortie, la définition des fonctions d'appartenance pour chaque terme linguistique et la dérivation des règles de contrôle [48]. Le choix d'une méthode de conception dépend étroitement de la nature du domaine de travail et de l'information disponible. Dû à la complexité de l'espace de recherche pour le problème de conception d'un contrôleur flou. Dû, entre autres, à la non dérivabilité et la discontinuité des fonctions impliquées, la seule alternative pour résoudre ce type de problème est l'utilisation des algorithmes génétiques.

L'incorporation de l'apprentissage génétique dans un processus de conception floue ajoute une dimension intelligente au FLC qui le permet de générer automatiquement ses paramètres optimaux (Figure 6.1). L'interaction des différents blocs d'un système de contrôle est illustrée dans l'organigramme de la figure 6.2.

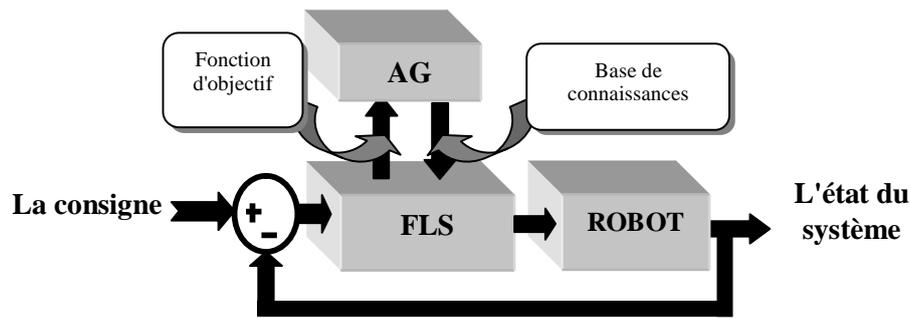


Figure 6.1: Structure du système de contrôle.

Concevoir un FLC consiste principalement à définir le ou les paramètres à optimiser, le type de codage des paramètres et la fonction coût.

6.3.1. Description du contrôleur flou proposé :

Le contrôleur proposé est basé sur la règle d'implication de MAMDANI (max-min), la conjonction floue (min) et par la méthode de centre de gravité pour la défuzzification. Il a comme entrées l'erreur et le changement d'erreur sur la position de l'articulation et fournit à sa sortie, le couple de commande à appliquer au robot pour ramener l'axe correspondant à la position désirée.

Les fonctions d'appartenance utilisées pour la partition floue de l'espace des variables d'entrées et de sortie sont triangulaires, symétriques et équidistantes. On a associé pour la variable erreur (E_r) trois termes : Positif (P), Zéro (Z) et Négatif (N) répartis sur l'univers de discours $[-0.05, 0.05]$, pour le changement d'erreur (Der) cinq termes sont associés : Négatif Grand (NG), Négatif Petit (NP), Zéro (Z), Positif Petit (PP) et Positif Grand (PG) répartis sur l'univers de discours $[-0.05, 0.05]$ et pour le couple de commande (C_p) sept paramètres sont associés : Négatif Grand (NG), Négatif Moyen (NM), Négatif Petit (NP), Zéro (Z), Positif Petit (PP), Positif Moyen (PM) et Positif Grand (PG) répartis sur l'univers de discours $[-5, 5]$.

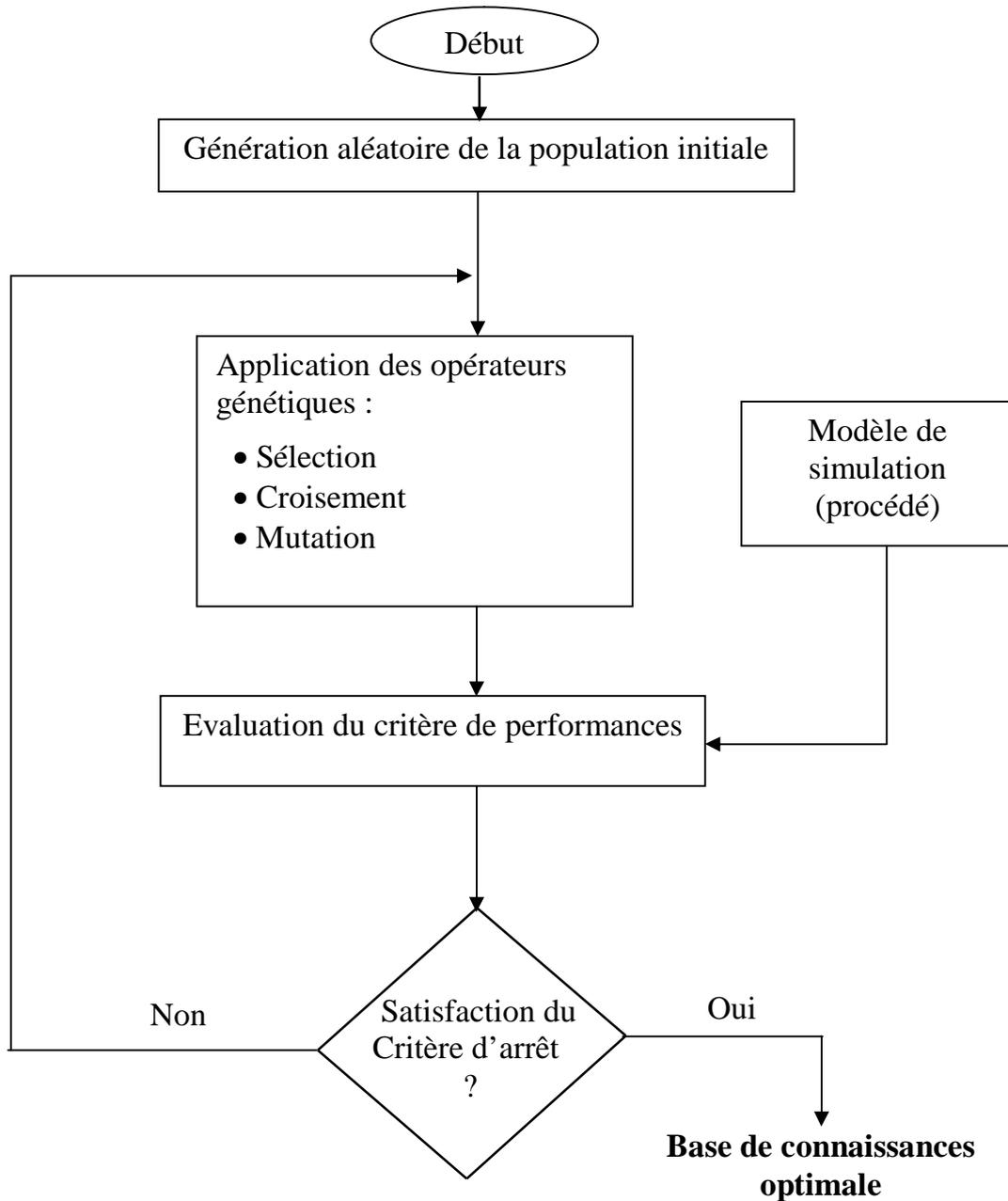


Figure 6.2: Etapes de conception d'un contrôleur flou.

6.3.2. Définition des paramètres à optimiser :

Il y a trois approches d'application des AGs pour la conception des contrôleurs flous. Dans la première, la base des règles (règles floues) est fixée, tandis que la base de données (définitions des fonctions d'appartenance) est optimisée par l'AG. Dans la seconde, la base de données est fixée seulement, la base des règles est optimisée par l'AG (détermination des règles floues). Dans la troisième approche toute la base de connaissances (base de données + base de règles) est optimisée simultanément pour avoir les meilleures performances.

C'est la troisième approche qui a été développée dans notre travail pour concevoir le système de contrôle de mouvement.

Les paramètres à optimiser dans cette application sont :

1- Les paramètres décrivant les fonctions d'appartenance du contrôleur flou, et puisque ces fonctions sont triangulaires, symétriques et équidistantes, un seul paramètre est nécessaire pour les décrire (Figure 6.3).

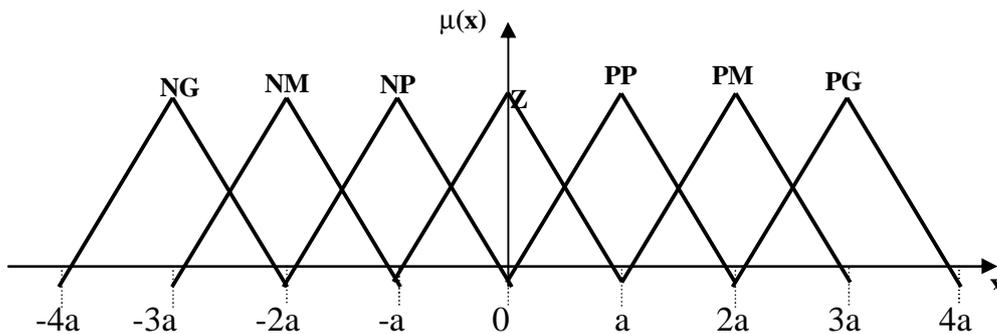


Figure 6.3: Les paramètres décrivant la fonction d'appartenance.

Afin de décrire les fonctions d'appartenance des trois variables d'entrées et de sortie, il nous faut trois paramètres à optimiser. Puisqu'il s'agit du contrôle d'un robot à deux degrés de liberté, nous avons besoin de deux contrôleurs flous pour chacun de ces axes, alors le nombre de paramètres à optimiser devient $3 \times 2 = 6$ paramètres.

2- La base de règles qui contient $(3 \times 5) \times 2 = 30$ règles floues doivent être optimisées par l'algorithme génétique.

6.3.3. Codage des paramètres :

Le codage utilisé est un codage binaire où plusieurs paramètres sont codés et juxtaposés les uns après les autres. Ces paramètres sont les 30 règles floues décrivant la base de règles et les 6 paramètres décrivant les fonctions d'appartenance des deux contrôleurs flous. Ils sont codés sur un chromosome de longueur de 120 allèles.

Les 90 premiers allèles représentent le codage de la base de règles, chaque règle est codée sur 3 bits ce qui donne $30 \times 3 = 90$ bits (Tableau 6.1). Le tableau 6.2 représente l'idée du codage binaire des règles floues sur 3 bits. Les 30 allèles restants sont réservés au codage des 6 paramètres décrivant les fonctions d'appartenance avec une précision de 5 bits par paramètre (Tableau 6.3).

Er \DEr		NG	NP	Z	PP	PG
FLC1	N	Chrom[1-3]	chrom[4-6]	Chrom[7-9]	chrom[10-12]	chrom[13-15]
	Z	Chrom[16-18]	chrom[19-21]	Chrom[22-24]	chrom[25-27]	chrom[28-30]
	P	Chrom[31-33]	chrom[34-36]	Chrom[37-39]	chrom[40-42]	chrom[43-45]
FLC2	N	Chrom[46-48]	chrom[49-51]	Chrom[52-54]	chrom[55-57]	chrom[58-60]
	Z	Chrom[61-63]	chrom[64-66]	Chrom[67-69]	chrom[70-72]	chrom[73-75]
	P	Chrom[76-78]	chrom[79-81]	Chrom[82-84]	chrom[85-87]	chrom[88-90]

Tableau 6.1: Codage de la base des règles floues

Ensemble flou	NG	NM	NP	Z	PP	PM	PG	Z
Codage	000	001	010	011	100	101	110	111

Tableau 6.2: Codage binaire des règles floues.

Variable		Variable flou	Paramètres	Codage
FLC1		Erreur	a_1	chrom[91-95]
		Changement d'erreur	a_2	chrom[96-100]
		Couple de commande	a_3	chrom[101-105]
FLC2		Erreur	a_4	chrom[106-110]
		Changement d'erreur	a_5	chrom[111-115]
		Couple de commande	a_6	chrom[116-120]

Tableau 6.3: Codage des paramètres des fonctions d'appartenance

Comme nous l'avons mentionné dans le chapitre 2 (équation 2.12), la précision du codage dépend essentiellement des bornes supérieure et inférieure du domaine de variation de la variable à coder et de la longueur de la chaîne codant cette variable.

$$\text{Précision} = \frac{U_{\max} - U_{\min}}{2^l - 1}.$$

Dans notre cas : $P_{Er} = 3.2 \times 10^{-3}$, $P_{Der} = 3.2 \times 10^{-3}$ et $P_{Cp} = 0.065$.

6.3.4. Fonction d'objectif :

L'AG a pour but de générer une solution optimale satisfaisant la minimisation de l'erreur quadratique en position et en vitesse et de la quantité d'énergie de commande à appliquer sur le robot, cette idée est exprimée par la relation suivante :

$$J = \sum_{k=1}^{150} \sqrt{c_1 \cdot e^2(k) + c_2 \cdot (\dot{e})^2(k) + c_3 \cdot \tau^2(k)} \quad (6.1)$$

Les coefficients de pondération c_1 , c_2 et c_3 sont choisis empiriquement pour éviter les débordements.

Puisque les AGs sont par nature des procédures de recherche de maximum, il faut transformer le problème de minimisation de J en un problème de maximisation de f donnée

par la relation suivante :

$$f = \frac{1}{1+J} \quad (6.2)$$

6.3.5. Les opérateurs génétiques :

L'AG proposé utilise la sélection par roue de loterie avec remplacement, un opérateur de croisement en deux points et un opérateur de mutation standard.

Les paramètres de l'AG utilisés sont :

- La taille de la population est fixée à 100.
- Le nombre maximal de génération est égal à 200.
- La probabilité de mutation est égale à 0.02.
- La probabilité de croisement est égale à 0.95.
- Les constantes de pondération c_1 , c_2 et c_3 valent : 10^{-1} , 5×10^{-1} et 10^{-3} respectivement.

Pour effectuer la simulation, un modèle décrivant le système à commander (robot manipulateur dans notre cas) est nécessaire, l'obtention de ce modèle nécessite une modélisation détaillée de la dynamique du système à commander.

6.4. Description et modélisation du robot manipulateur :

Pour effectuer l'analyse et la synthèse d'un système dynamique, il est nécessaire de connaître les relations entre ses grandeurs d'entrées et de sorties. L'ensemble de ces relations constitue le modèle mathématique du système, l'efficacité de ce modèle repose sur une analogie entre le comportement des objets physiques et celui des êtres mathématiques.

6.4.1 Description du robot manipulateur :

Nous avons opté pour cette application, un robot manipulateur de type PLANAR, sa structure mécanique est à chaîne simple ouverte caractérisée par deux articulations rotoïdes dont les mouvements sont identifiés par les variables θ_1 et θ_2 (Figure 6.4).

Les paramètres mécaniques du robot étudié sont donnés dans le tableau 6.4.

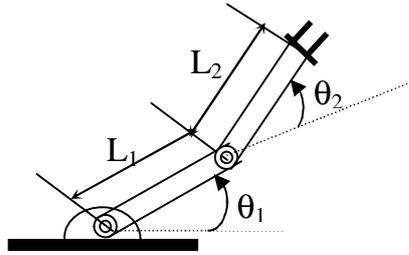


Figure 6.4: Structure du robot à 2 d.d.l.

Paramètres	Notation	Axe ₁	Axe ₂	Unité
Longueur du segment	L	1.0	1.0	m
masse du segment	m	1	0.6	kg

Tableau 6.4: Paramètres mécaniques du robot

6.4.2. Modélisation géométrique du robot :

La figure 6.5 montre le placement des repères associés aux articulations du robot. Le sens de rotation des deux joints est choisi autour de l'axe z qui est perpendiculaire à la surface de déplacement du robot.

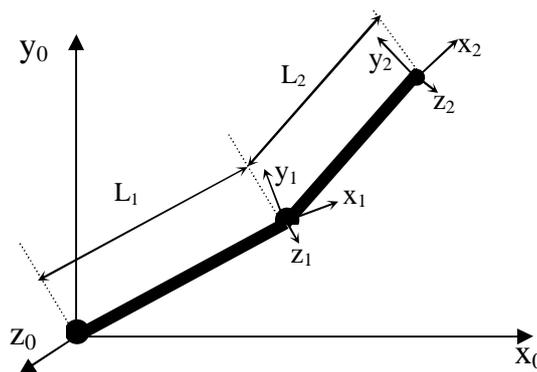


Figure 6.5: Placement des repères sur les joints

6.4.2.1. Modèle géométrique direct (MGD) :

C'est l'ensemble des relations qui permettent d'exprimer la situation de l'organe terminal du robot en fonction de ses coordonnées articulaires.

La situation de l'organe terminal est définie par :

$$P = [x \ y \ z]^T \quad (6.3)$$

Les variables articulaires sont définies par :

$$q = [\theta_1 \ \theta_2]^T \quad (6.4)$$

Les matrices de transformations homogènes élémentaires décrivant le comportement géométrique du robot PLANAR sont les suivantes (Equation 2. 5) :

Paramètres de D.H.	d	a	θ	α
Axe1	0.0	L ₁	θ_1	0.0
Axe2	0.0	L ₂	θ_2	0.0

Tableau 6.5: Les paramètres de D.H du robot PLANAR

$${}^0T_1 = \begin{bmatrix} c_1 & -s_1 & 0 & L_1c_1 \\ s_1 & c_1 & 0 & L_1s_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^1T_2 = \begin{bmatrix} c_2 & -s_2 & 0 & L_2c_2 \\ s_2 & c_2 & 0 & L_2s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Afin de calculer la matrice de transformation globale, on doit faire la multiplication des matrices élémentaires :

$${}^0T_2 = {}^0T_1 \times {}^1T_2 = \begin{bmatrix} c_{12} & -s_{12} & 0 & L_2c_{12} + L_1c_1 \\ s_{12} & c_{12} & 0 & L_1s_{12} + L_1s_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Avec : $c_i = \cos\theta_i$, $s_i = \sin\theta_i$, $c_{ij} = \cos(\theta_i + \theta_j)$ et $s_{ij} = \sin(\theta_i + \theta_j)$.

Le modèle géométrique final est donné par les relations :

$$\begin{cases} x = L_2c_{12} + L_1c_1 \\ y = L_2s_{12} + L_1s_1 \\ z = 0 \end{cases} \quad (6.5)$$

La valeur de z est toujours nulle, c'est pourquoi ce robot est dit PLANAR.

6.4.2.2. Modèle géométrique inverse (MGI) :

C'est l'inverse du problème précédent, il consiste à calculer les coordonnées articulaires en fonction de celles opérationnelles. Ce modèle est donné par la relation :

$$\mathbf{q} = \mathbf{f}^{-1}(\mathbf{P}) \quad (6.6)$$

Le modèle géométrique inverse du robot PLANAR est donné par les relations suivantes :

$$\begin{cases} \theta_1 = \arctan 2(y(L_2 c_2 + L_1) - L_2 x s_{12}, y L_2 s_{12} + x(L_2 c_2 + L_1)) \\ \theta_2 = \arctan 2((x^2 + y^2 - L_1^2 - L_2^2)/(2L_1 L_2), \sqrt{1 - c_2^2}) \end{cases} \quad (6.7)$$

6.4.3. Modélisation dynamique du robot :

Ce modèle décrit la relation entre les couples (et/ou forces) appliqués aux actionneurs et les positions, vitesses et accélérations articulaires. Il s'écrit sous la forme (Equation 2.16) :

$$\Gamma = \mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \mathbf{F}) \quad (6.8)$$

Pour déterminer ce modèle, il existe plusieurs formalismes, mais pour notre cas nous préférons le formalisme de LAGRANGE qui décrit les équations de mouvement en terme de travail et énergie du système :

$$\tau_i = d(\partial L / \partial \dot{q}_i) / dt - \partial L / \partial q_i \quad (6.9)$$

A partir de la définition de la matrice Q qui est donnée pour une articulation rotoïde par :

$$Q_i = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

On calcule U_{ij} , à partir de l'équation (2.27) :

$$U_{11} = \frac{\partial^0 T_1}{\partial \theta_1} = Q_1 \cdot {}^0 T_1 = \begin{bmatrix} -s_1 & -c_1 & 0 & -L_1 s_1 \\ c_1 & -s_1 & 0 & L_1 c_1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$U_{21} = \frac{\partial {}^0T_2}{\partial \theta_1} = Q_1 \cdot {}^0T_2 = \begin{bmatrix} -s_{12} & -c_{12} & 0 & -(L_1 s_1 + L_2 s_{12}) \\ c_{12} & -s_{12} & 0 & (L_1 c_1 + L_2 c_{12}) \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$U_{22} = \frac{\partial {}^0T_2}{\partial \theta_2} = {}^0T_1 \cdot Q_2 \cdot {}^1T_2 = \begin{bmatrix} -s_{12} & -c_{12} & 0 & -L_2 s_{12} \\ c_{12} & -s_{12} & 0 & L_2 c_{12} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Les matrices d'inerties du robot sont données par (Equation 2.35) :

$$J_1 = \begin{bmatrix} \frac{1}{3}m_1L_1^2 & 0 & 0 & -\frac{1}{2}m_1L_1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -\frac{1}{2}m_1L_1 & 0 & 0 & m_1 \end{bmatrix}, J_2 = \begin{bmatrix} \frac{1}{3}m_2L_1^2 & 0 & 0 & \frac{1}{2}m_2L_2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{2}m_2L_2 & 0 & 0 & m_2 \end{bmatrix}$$

En utilisant l'équation (2.47), les éléments de la matrice D seront :

$$D_{11} = \text{Tr}(U_{11} \cdot J_1 \cdot U_{11}^T) + \text{Tr}(U_{21} \cdot J_2 \cdot U_{21}^T)$$

$$= \frac{1}{3}m_1L_1^2 + \frac{1}{3}m_2L_2^2 + m_2L_1^2 + m_2L_1^2 \cos\theta_2 \quad (6.10)$$

$$D_{12} = D_{21} = \text{Tr}(U_{22} \cdot J_2 \cdot U_{21}^T)$$

$$= \frac{1}{3}m_2L_2^2 + \frac{1}{2}m_2L_1L_2 \cos\theta_2 \quad (6.11)$$

$$D_{22} = \text{Tr}(U_{22} \cdot J_2 \cdot U_{22}^T)$$

$$= \frac{1}{3}m_2L_2^2 \quad (6.12)$$

Pour calculer les termes de coriolis et de centrifuge, on utilise l'équation (2.48) :

$$h_1 = \sum_{k=1}^2 \sum_{m=1}^2 h_{1km} \dot{\theta}_k \dot{\theta}_m$$

$$= -\frac{1}{2}m_2L_1L_2 \sin\theta_2 \dot{\theta}_2^2 - m_2L_1L_2 \sin\theta_2 \dot{\theta}_1 \dot{\theta}_2 \quad (6.13)$$

$$\begin{aligned} h_2 &= \sum_{k=1}^2 \sum_{m=1}^2 h_{2km} \dot{\theta}_k \dot{\theta}_m \\ &= \frac{1}{2}m_2L_1L_2 \sin\theta_2 \dot{\theta}_1^2 \end{aligned} \quad (6.14)$$

Ensuite, on a besoin de dériver les éléments du vecteur de gravité C à partir de l'équation (Equation 2.50) :

$$\begin{aligned} C_1 &= -(m_1GU_{11}^1 \bar{r}_1 + m_2GU_{21}^2 \bar{r}_2) \\ &= \frac{1}{2}m_1gL_1 \cos\theta_1 + \frac{1}{2}m_2gL_2 \cos(\theta_1 + \theta_2) + m_2gL_1 \cos\theta_1 \end{aligned} \quad (6.15)$$

$$\begin{aligned} C_2 &= -m_2GU_{22}^2 \bar{r}_2 \\ &= \frac{1}{2}m_2gL_2 \cos(\theta_1 + \theta_2) \end{aligned} \quad (6.16)$$

Finalement, l'équation de mouvement du robot PLANAR trouvée est donnée sous forme matricielle par la relation :

$$\tau(t) = D(\theta) \ddot{\theta}(t) + h(\theta, \dot{\theta}) + C(\theta) \quad (6.17).$$

$$\begin{aligned} \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} &= \begin{bmatrix} \frac{1}{3}m_1L_1^2 + \frac{1}{3}m_2L_2^2 + m_2L_1^2 + m_2L_1L_2 \cos\theta_2 & \frac{1}{3}m_2L_2^2 + \frac{1}{2}m_2L_1L_2 \cos\theta_2 \\ \frac{1}{3}m_2L_2^2 + \frac{1}{2}m_2L_1L_2 \cos\theta_2 & \frac{1}{3}m_2L_2^2 \end{bmatrix} \times \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} \\ &+ \begin{bmatrix} -\frac{1}{2}m_2L_1L_2 \sin\theta_2 \dot{\theta}_2^2 - m_2L_1L_2 \sin\theta_2 \dot{\theta}_1 \dot{\theta}_2 \\ \frac{1}{2}m_2L_1L_2 \sin\theta_2 \dot{\theta}_1^2 \end{bmatrix} \\ &+ \begin{bmatrix} \frac{1}{2}m_1gL_1 \cos\theta_1 + \frac{1}{2}m_2gL_2 \cos(\theta_1 + \theta_2) + m_2gL_1 \cos\theta_1 \\ \frac{1}{2}m_2gL_2 \cos(\theta_1 + \theta_2) \end{bmatrix} \end{aligned} \quad (6.18)$$

La simulation du modèle de connaissances établi par le formalisme de LAGRANGE nécessite la mise sous forme d'état des équations différentielles non linéaires trouvées (Equation 6.18), la forme d'état établie est [65] :

Notons :

$$\left\{ \begin{array}{l} \dot{X}_1 = \ddot{\theta}_1 \\ \dot{X}_2 = \ddot{\theta}_2 \\ \dot{X}_3 = \dot{\theta}_1 \\ \dot{X}_4 = \dot{\theta}_2 \end{array} \right. , \text{ et donc : } \left\{ \begin{array}{l} \theta_1 = X_3 \\ \theta_2 = X_4 \\ \dot{\theta}_1 = X_1 \\ \dot{\theta}_2 = X_2 \\ \ddot{\theta}_1 = \dot{X}_1 \\ \ddot{\theta}_2 = \dot{X}_2 \end{array} \right. \quad (6.19)$$

En substituant l'équation (6.19) dans l'équation (6.18), on obtient un système différentiel de premier ordre donné par l'expression suivante :

$$\begin{aligned} \begin{bmatrix} \dot{X}_1 \\ \dot{X}_2 \end{bmatrix} &= \begin{bmatrix} \frac{1}{3}m_1L_1^2 + \frac{1}{3}m_2L_2^2 + m_2L_1^2 + m_2L_1L_2 \cos(X_4) & \frac{1}{3}m_2L_2^2 + \frac{1}{2}m_2L_1L_2 \cos(X_4) \\ \frac{1}{3}m_2L_2^2 + \frac{1}{2}m_2L_1L_2 \cos(X_4) & \frac{1}{3}m_2L_2^2 \end{bmatrix}^{-1} \\ &\times \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} - \begin{bmatrix} -\frac{1}{2}m_2L_1L_2 \sin(X_4)X_3^2 - m_2L_1L_2 \sin(X_4)X_1 \cdot X_2 \\ \frac{1}{2}m_2L_1L_2 \sin(X_4)X_1^2 \end{bmatrix} \\ &- \begin{bmatrix} \frac{1}{2}m_1gL_1 \cos(X_3) + \frac{1}{2}m_2gL_2 \cos(X_3 + X_4) + m_2gL_1 \cos(X_3) \\ \frac{1}{2}m_2gL_2 \cos(X_3 + X_4) \end{bmatrix} \quad (6.20) \end{aligned}$$

La méthode de RANGE-KUTTA d'ordre quatre est sélectionnée pour résoudre numériquement l'équation (6.20), avec un choix judicieux du pas de calcul $dt = 0.01s$.

6.5. Résultats de simulation :

Dans cette section, nous allons étudier le contrôle de poursuite du robot manipulateur décrit dans le paragraphe (6.3.2.) par le système de contrôle flou proposé.

La trajectoire à poursuivre par l'organe terminal du robot est de forme elliptique (Figure 6.6), elle est donnée dans l'espace cartésien par l'équation (6.21).

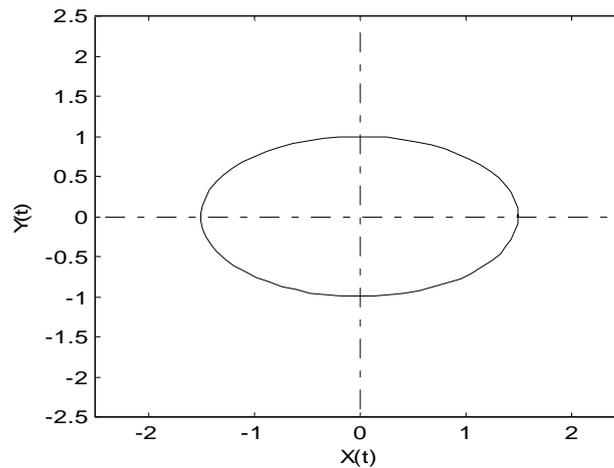


Figure 6.6: Trajectoire à poursuivre par l'organe terminal.

Cette trajectoire se traduit dans l'espace articulaire par les courbes représentées sur la figure 6.7.

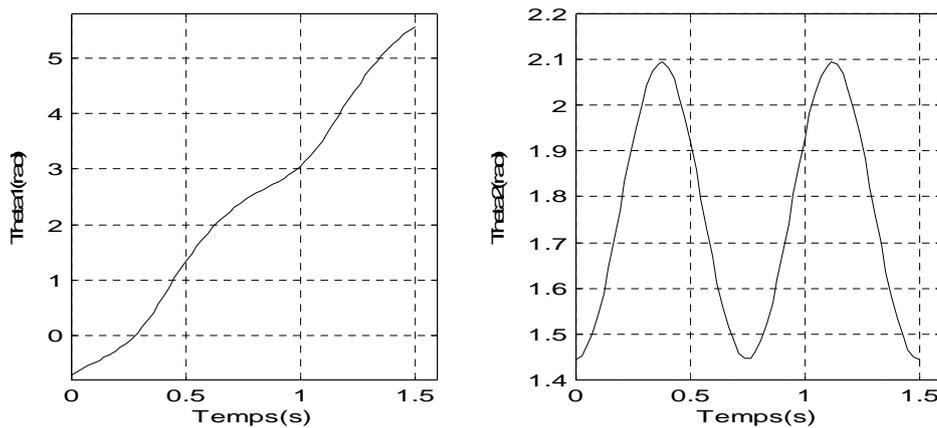


Figure 6.7: Trajectoires à poursuivre par chacune des articulations.

$$\begin{cases} X(t)=1.5\cos(4.2t) \\ Y(t)=1.\sin(4.2t) \end{cases} \quad (6.21)$$

L'équation 6.21 représente une trajectoire elliptique, elle est, d'après M. Simon X. Yang et Max Meng [69], la trajectoire la plus difficile à poursuivre par le robot PLANAR. Ceci va nous permettre d'évaluer effectivement les performances du contrôleur proposé.

L'évolution de la fonction d'objectif en fonction du nombre de générations, représentée sur la figure 6.8, est le meilleur résultat en terme de valeur finale de la fonction d'objectif obtenu après 3 exécutions de l'algorithme génétique.

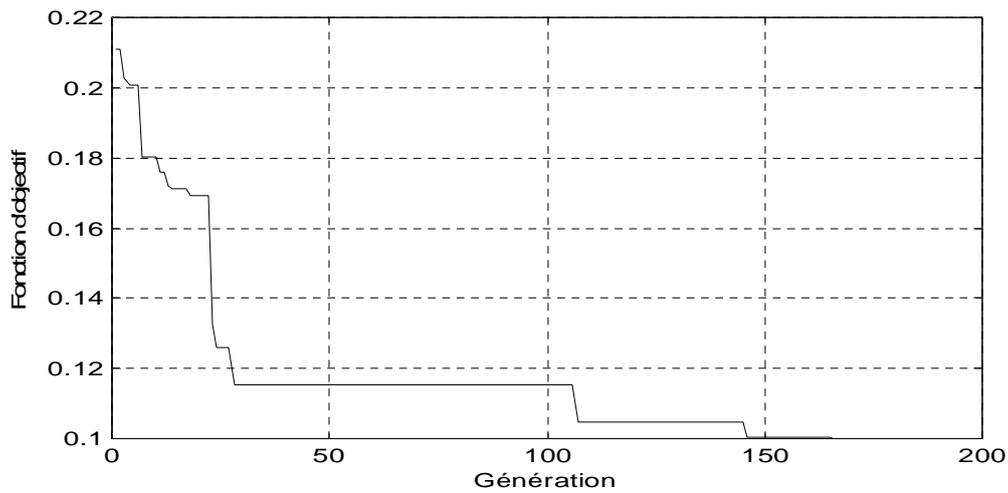


Figure 6.8: Evolution de la fonction d'objectif.

Le tableau suivant donne les valeurs des six (6) paramètres décrivant les fonctions d'appartenance des variables d'entrées et de sorties, ces paramètres représentent le résultat d'exécution de l'algorithme génétique proposé.

Paramètre	a_1	a_2	a_3	a_4	a_5	a_6
Code	00111	00101	10001	00110	00011	01000
Valeur	0.022	0.016	1.1	0.02	0.01	0.51

Tableau 6.6: Valeurs des paramètres optimisés par L'AG.

Les bases de connaissances des deux contrôleurs flous optimisées par l'AG sont représentées sur les figures 6. 9-14.

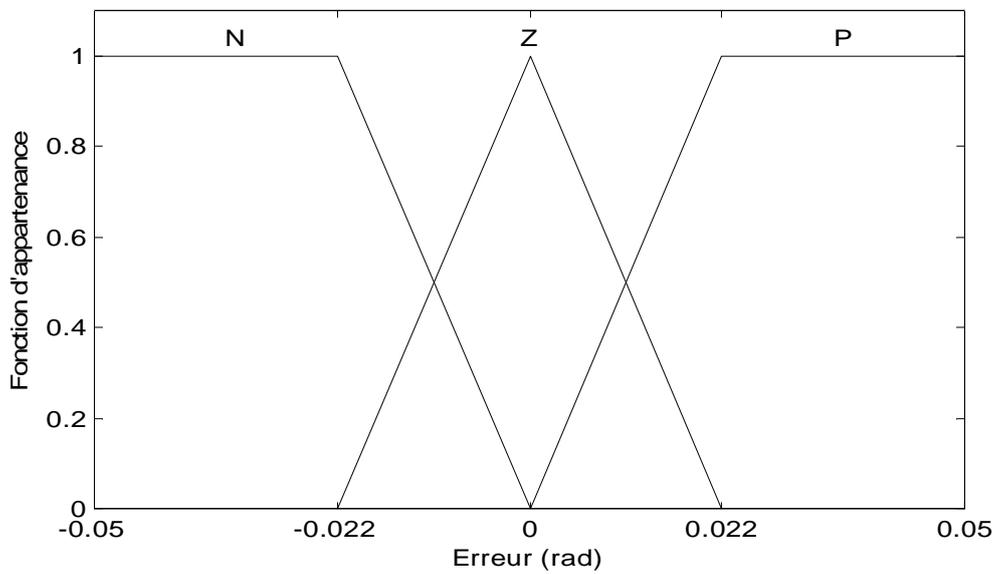


Figure 6.9: Fonctions d'appartenance de la variable Er1

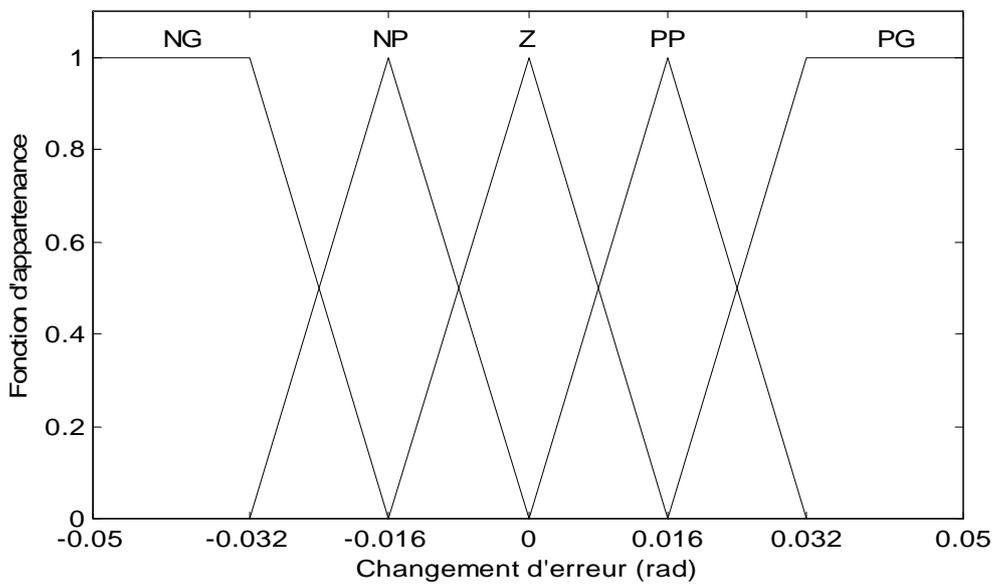


Figure 6.10: Fonctions d'appartenance de la variable Der1.

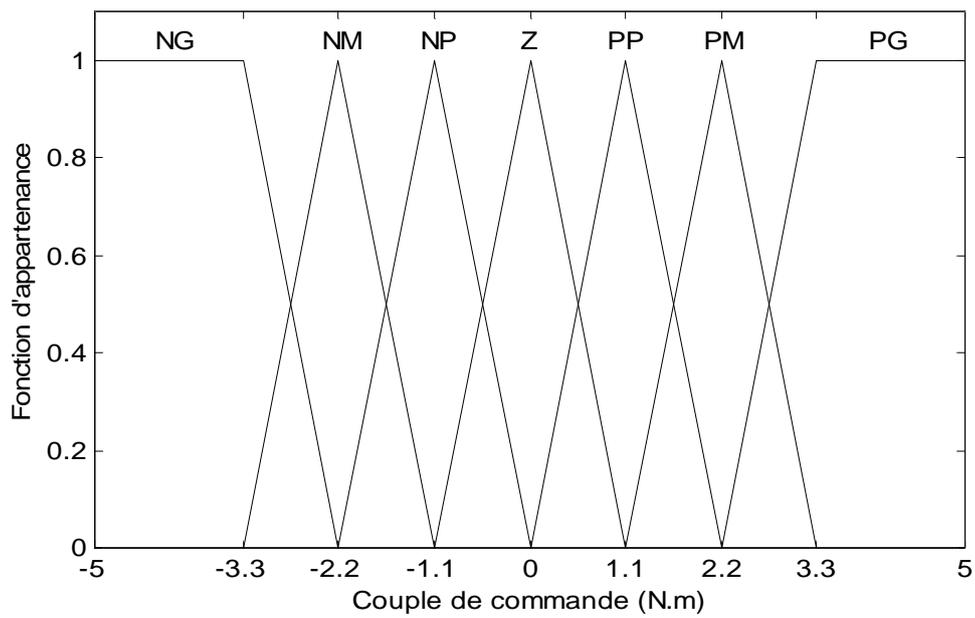


Figure 6.11: Fonctions d'appartenance de la variable Cp1.

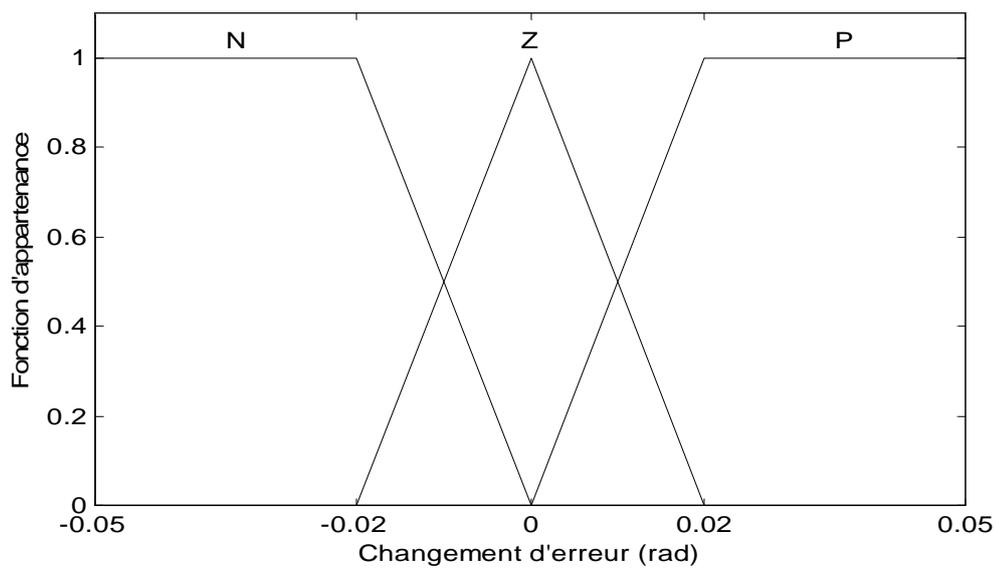


Figure 6.12: Fonctions d'appartenance de la variable Er2.

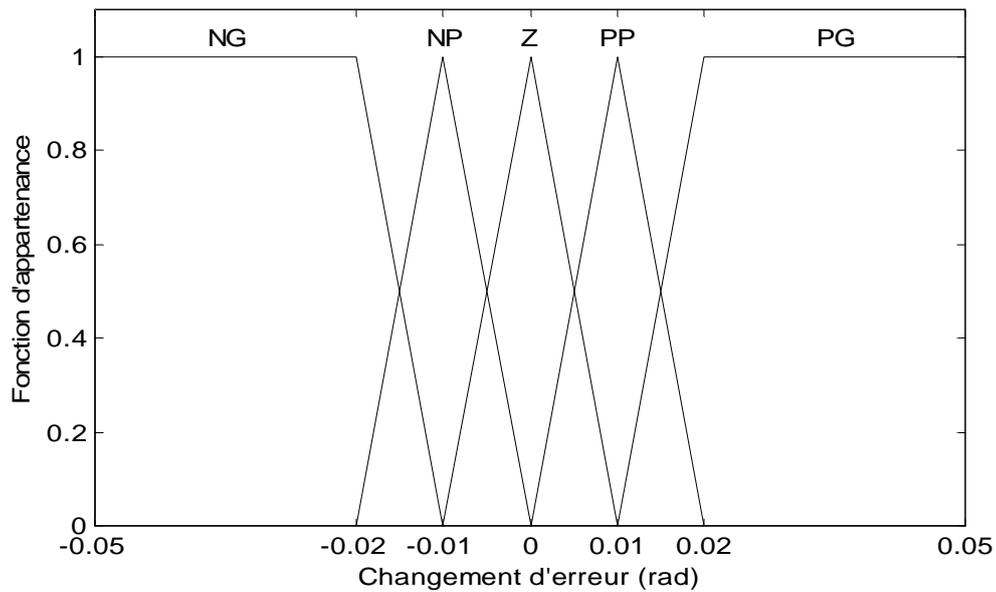


Figure 6.13: Fonctions d'appartenance de la variable Der2

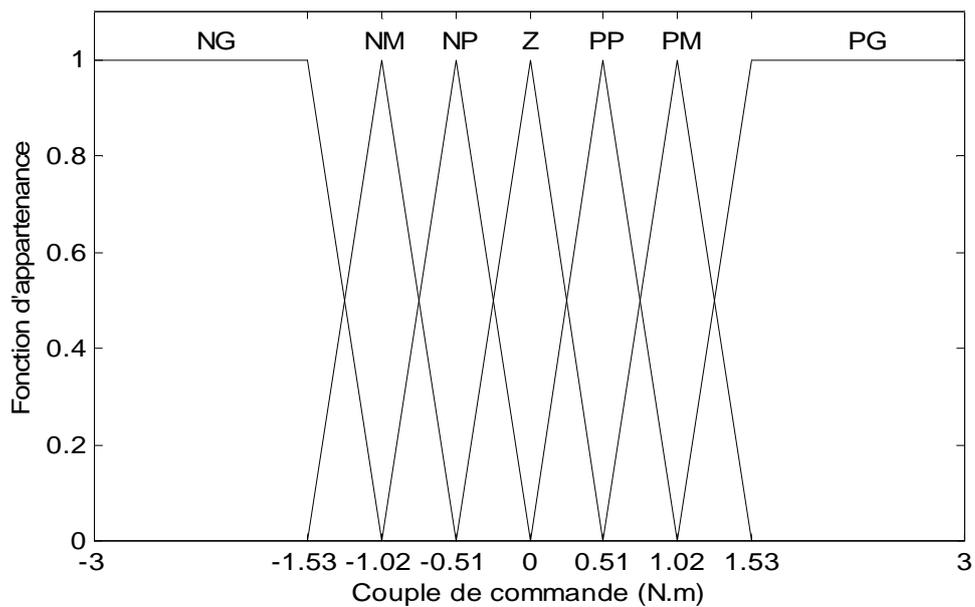


Figure 6.14: Fonctions d'appartenance de la variable Cp2.

Les bases des règles des deux contrôleurs sont données dans les tableaux suivants :

Er1\Der1	NG	NP	Z	PP	PG
N	Z	NP	NM	NM	NG
Z	NM	NP	Z	PP	PM
P	Z	PP	PM	PM	PG

Tableau 6.7: Base de règles floues du premier contrôleur

Er2\Der2	NG	NP	Z	PP	PG
N	PP	Z	NM	NM	NG
Z	NG	NM	Z	PP	PM
P	NP	Z	PP	PM	PG

Tableau 6.8: Base de règles floues du deuxième contrôleur.

Les performances de poursuite des signaux de référence, ainsi que les erreurs de poursuite sont illustrées dans les figures 6.15-17 :

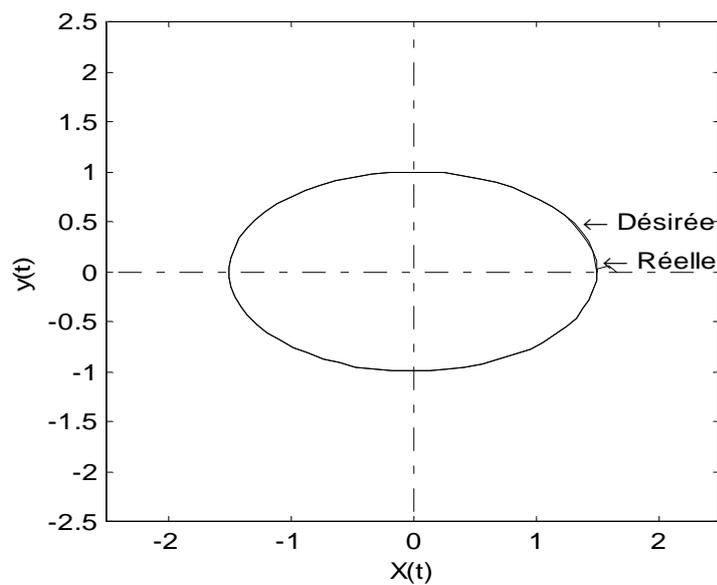


Figure 6.15: Performances de poursuite dans l'espace cartésien

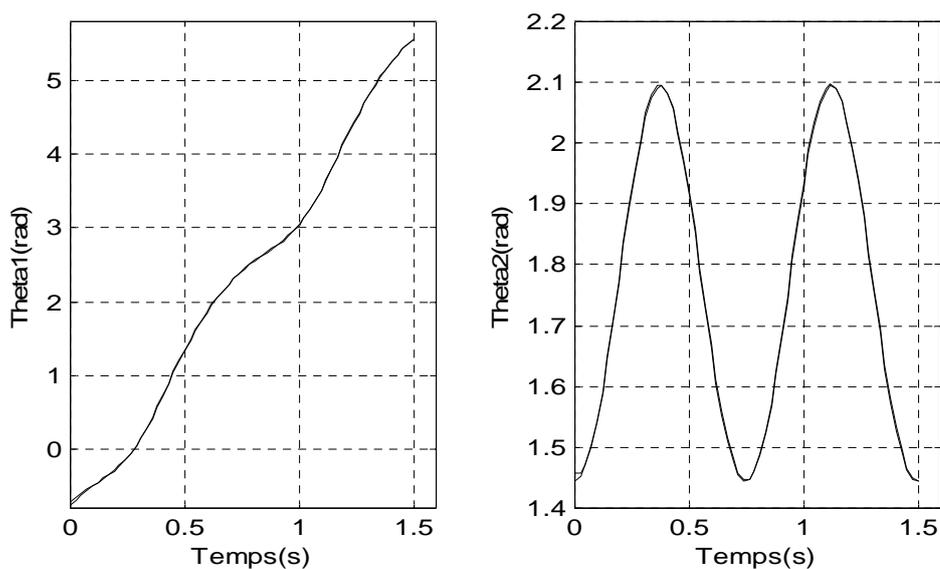


Figure 6.16: Performances de poursuite dans l'espace articulaire.

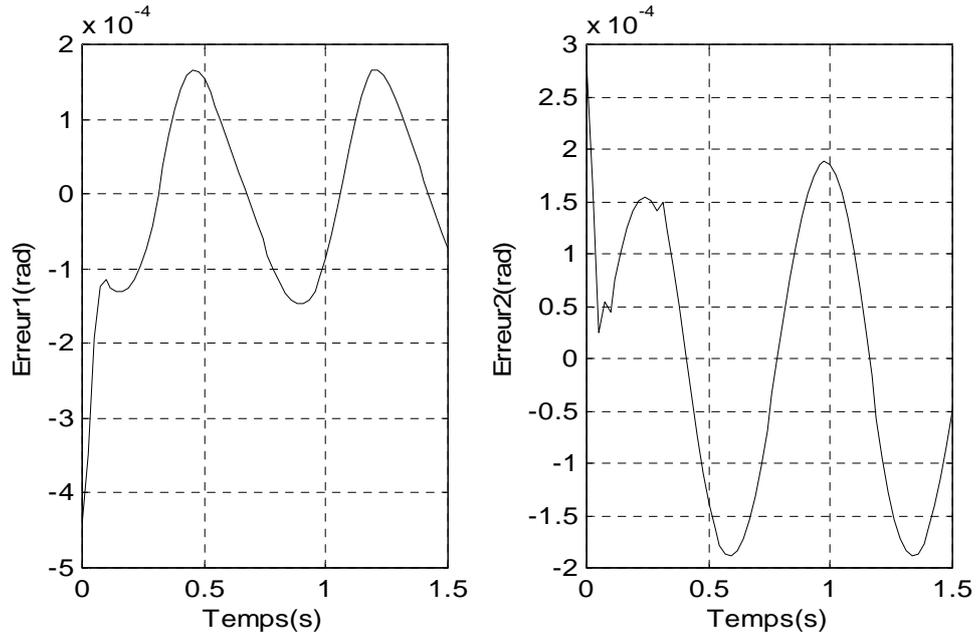


Figure 6.17: Erreurs de poursuite dans l'espace articulaire .

Pour tester la robustesse du système flou proposé face aux variations paramétriques et aux perturbations externes, nous avons augmenté la masse du deuxième segment du robot de 50% de sa masse initiale à l'instant $t = 0.4\text{sec}$.

Les résultats de simulation illustrant cette situation sont représentés sur les figures (6.18-20).

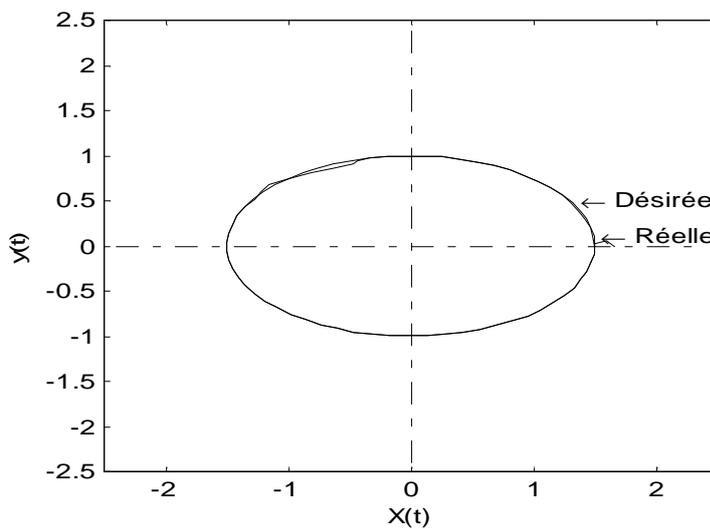


Figure 6.18: Performances de poursuite dans l'espace cartésien sous des conditions sévères.

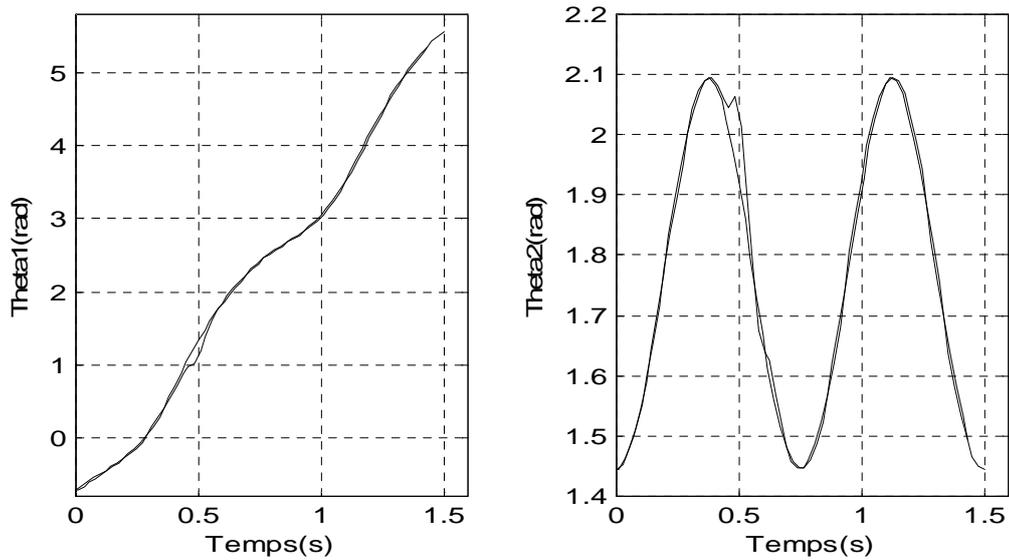


Figure 6.19: Performances de poursuite dans l'espace articulaire sous des conditions sévères.

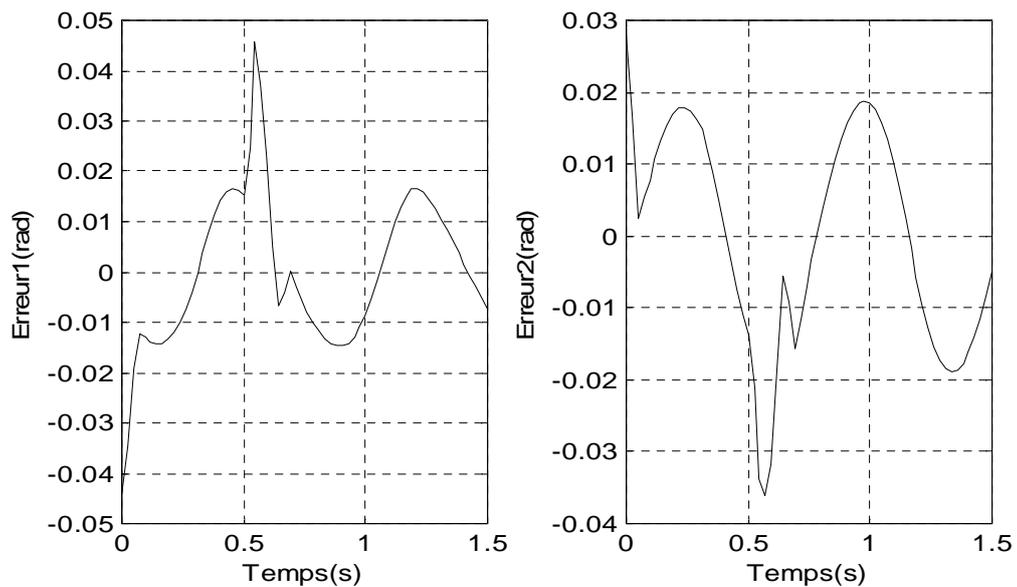


Figure 6.20: Erreurs de poursuite dans l'espace articulaire sous des conditions sévères.

Les résultats de simulation, précédemment illustrés, montrent clairement que les performances de poursuite et la robustesse du contrôleur proposé sont très bonnes même sous des conditions sévères.

6.6. Conclusion :

Dans ce chapitre, un système de contrôle de mouvement des robots manipulateurs a été développé. Ce système est basé sur deux approches originales à savoir, la logique floue et les algorithmes génétiques. L'utilisation du contrôle flou a éliminé le besoin du modèle mathématique exact, et l'utilisation de l'algorithme génétique a éliminé le besoin d'un expert humain dans la phase de conception de tel contrôleur. Le système de contrôle flou ainsi développé a été appliqué à la commande décentralisée d'un bras manipulateur à 2 degrés de liberté de type PLANAR.

Les résultats de simulation, envisagés ici, montrent l'efficacité de la technique de conception des contrôleurs flous par Les AGs pour accomplir un contrôle de poursuite précis et robuste des robots manipulateurs.

Conclusion générale

La commande des manipulateurs mobiles reste jusqu'à présent un thème de recherche important. Bien que d'innombrables travaux dans ce domaine sont apparus, des solutions globales pour de tels systèmes demeurent toujours posées en raison de leurs redondances et de la difficulté de leur modélisation en tenant compte des aspects de la stabilité du système, de sa vitesse d'exécution, de la coopération et de la coordination. Une synthèse des travaux réalisés, que nous avons présenté met le point sur les différents travaux concernant la locomotion, la manipulation et la génération du mouvement.

Dans cette thèse nous nous sommes intéressés à développer un système de planification et de contrôle de mouvement d'un manipulateur mobile. L'originalité dans ce travail réside dans l'intégration de la logique floue et des algorithmes génétiques dans le processus de commande du système de la manipulation mobile. La méthode repose sur l'avantage de la commande séparée des deux sous-systèmes (la plate-forme mobile et le bras manipulateur). On résout de cette manière le problème de complexité, qui nécessite autrement la modélisation du système global, en lui donnant plus de flexibilité dans la gestion des tâches dans des milieux contraignants.

La commande du bras manipulateur est résolu par un planificateur de mouvement conçu par un système composé de deux blocs, le premier bloc est un planificateur de mouvement, il sert à générer la trajectoire que le robot doit suivre pour déplacer son organe terminal de la position initiale à la position finale désirée dans un temps minimal et sans collision avec les obstacles. Afin de résoudre ce problème, un algorithme génétique a été utilisé pour optimiser la séquence de variations des angles des articulations sous la contrainte qu'aucune partie du robot n'entre en collision avec les obstacles existant dans son espace de travail. L'introduction de cette contrainte est faite par la méthode de pénalisation qui transforme le problème d'optimisation contraint en un problème non-contraint en associant un coût ou une pénalité à chaque violation de contrainte. Ce coût est inclus dans l'évaluation de la fonction d'objectif pour refléter à la fois la qualité et l'acceptabilité des solutions.

Le deuxième bloc est un contrôleur de mouvement chargé de contrôler le mouvement du robot de façon à ce que son organe terminal puisse suivre fidèlement la trajectoire générée par le bloc de planification de mouvement, pour ceci nous avons utilisé un contrôleur flou optimisé par un algorithme génétique dans lequel l'AG est utilisé pour développer les bases de connaissances des contrôleurs flous associés aux articulations du robot manipulateur. L'utilisation du contrôle flou a éliminé le besoin du modèle mathématique exact, et l'utilisation des AGs a éliminé le besoin d'un expert humain pour la conception des contrôleurs flous.

L'étude envisagée ici, montre l'efficacité de la technique de conception des contrôleurs flous par les algorithmes génétiques pour accomplir un contrôle de poursuite rapide, précis et robuste dans des environnements de travail encombrés.

Notre travail est basé essentiellement sur l'utilisation des algorithmes génétiques et malheureusement ces algorithmes ne sont pas bien adaptés au calcul en temps réel, cependant la planification et le contrôle de mouvement des manipulateurs mobiles nécessitent dans quelques cas le calcul en temps réel, notamment dans les environnements non stationnaires (dynamiques) qui contiennent des obstacles mobiles, ces obstacles peuvent être des robots travaillant dans le même espace de travail. Dans ce cas, pour que le manipulateur mobile puisse exécuter sa tâche, il doit générer en temps réel ses trajectoires avec une prise en compte instantanée des informations délivrées par les capteurs extéroceptifs. Aussi les bases de connaissances des contrôleurs flous doivent être modifiées en ligne pour pouvoir s'adapter à n'importe quelle trajectoire générée par le planificateur de mouvement.

A cet effet, nous proposons une direction très intéressante qui consiste à l'optimisation en temps réel des fonctions d'objectifs en travaillant sur un horizon d'erreur fini et peut-être sur quelques générations, ce qui permettra de réduire considérablement le temps mis par les AGs à converger vers la bonne solution en utilisant aussi d'autres approches d'aide à la convergence.

Nous envisageons ensuite la commande du manipulateur mobile en vu d'un déplacement d'une configuration initiale vers une configuration finale dans un environnement contraignant, en projetant deux modes de fonctionnement:

- Pour le premier mode, la base mobile sera utilisée pour transporter le bras manipulateur vers son espace de travail.
- Pour le deuxième, la base mobile et le bras manipulateur travaillent en coopération pour l'exécution d'une tâche continue dans l'espace de travail.

Pour formaliser le processus de raisonnement imprécis, on aura recours une fois de plus à l'application de la logique floue et à un algorithme génétique pour extraire l'action adéquate au bon suivi de trajectoires.

Comme nous penserons aussi appliquer les mêmes méthodes de commande pour le contrôle de la plate-forme et d'une structure mobile (Robot mobile) c'est-à-dire le manipulateur mobile ne sera plus considéré comme l'assemblage du bras et de la plate-forme mais sera pris plutôt comme un système unique. Le comportement de l'ensemble tiendra en considération les interactions dynamiques, la coordination des mouvements et la coopération.

Références bibliographiques

- [1] M. Zhao, N. Ansari, and E.S.H.Hou, " Mobile manipulator path planning by a genetic algorithm", *Journal of Robotic Systems*, 11(3): 143-153, 1994.
- [2] F. G. Pin and J. C. Culioli, " Optimal Positioning of Combined Mobile Platform-Manipulator systems for Material Handling Tasks", *Journal of intelligent and Robotic Systems*. 6: 165-182, 1992.
- [3] F. G. Pin, K. A. Morgansen, F. A. Tulloch, C. J. Hacker, and K. B. Gower, "Motion Planning for Mobile Manipulators with a Non-Holonomic Constraint Using the FSP (Full Space Parameterization) Method, *Journal of robotic Systems* 13(11), 723-736 ,1996.
- [4] J. K. Lee and H. S. Cho, " Mobile manipulator Motion Planning for Multiple Tasks Using Global Optimisation Approach", *Journal of Intelligent and Robotic Systems*, 18: 169-190, 1997.
- [5] H. Seraji, " Configuration control of rover-mounted manipulators ", *IEEE Int.Conf. on Robotics and Automation*, pp2261-2266, 1995.
- [6] S. Quinlan and O. Khatib, " Time path modification of collision-free paths ", In *Proc. of the IEEE Int.Conf. on Robotics and Automation*, vol. 2, pp802-807, 1993.
- [7] O. Brock and O. Khatib, " Real time replanning in high-dimensional configuration spaces using sets of homotopic paths", In *Proc. of the Intl.Conf. on Robotics and Automation*, vol. 1, pp550-555, 2000.
- [8] O. Brock and O. Khatib, Sriram Viji, " Task-Consistent Obstacle Avoidance and Motion Behavior for Mobile Manipulator ", In *Proc. of the IEEE Int.Conf. on Robotics and Automation*, 2002.
- [9] U.M. Nassal, " Motion coordination and reactive control of autonomous multi manipulators systems ", *Jour. of Robotic Systems*, 13(11):737-754, 1996.
- [10] Y.Yamamoto and X.Yun: " Coordination locomotion and manipulation of a mobile manipulator," *IEEE Transactions on Automatic Control*, vo139, no.6, pp 1326-1332, 1994.
- [11] T. Yoshikawa, " *Foundation of robotics: Analysis and Control*", The MIT Press. Cambridge, MA. 1990.

- [12] K. A. Tahboub, " Robust Control of Mobile Manipulators", *Journal of Robotic System*, 13(11) , 699-708,1996.
- [13] O. Khatib, 'Mobile Robotic Manipulation ', Symposium of the VDI/VDE society for Measurement and Automation Intelligent robot Control, Langen, Germany, Nov., pp 51-66,1993.
- [14]. O. Khatib, K. Yokoi, K.Chang, D. Ruspini, R. Holmberg, and A. Casai, 'Vehicle/arm Coordination and Multiple Mobile Manipulator Decentralised Cooperation ', Proc. IROS96, pp 546-553,1996.
- [15] R.B. Mc Ghee and A.A. frank, "On the Stability Properties of Quadruped Creeping Gait", *Mathematical Biosciences*, vol.3, No 2, pp 331-351, 1968.
- [16] Z.M. Huang and K.J. Waldron, "Relationship Between Payload Speed in Legged Locomotion System ", *IEEE Transactions on Robotics and Automation* , vol.6, No 5, pp 570-577, 1993.
- [17] D.Mesuri and C.A. Klein "Automac Body Regulation For Maintaining Stability of a Legged Vehicle During Routh-Terrain Locomotion ", *IEEE .J on Robotics and Automation* , vol. RA-1, No 3, pp 132-141, 1985.
- [18] A.Ghasempoor and N. Sepheri, "A measure of Machine Stability for Moving Base Manipulators ", *IEEE Inter .Conf. on Robotics and Automation*, pp 2249-2254,1995.
- [19] Q.Huang, K.Tanic and S.Sugano,," Coordinated Motion Planning for Mobile Manipulator Considering Stability and Manipulation", *The Inter.Journal of Robotics Research* , Vol. 19, No. 8, pp. 732-742, August 2000.
- [20] Yoshikawa, T."Manipulability of robotic mechanisms" *International Journal of Robotics Research (IJRR)*, 4(2) :3-9, 1985.
- [21] Foulon, G., Fourquet, J.-Y., et Renaud, M. "On coordinated tasks for nonholonomic mobile manipulators". Dans 5th IFAC Symp. Robot Control, pages 491-498, 1997.
- [22] Bayle, B., Fourquet, J.-Y., et Renaud, M. "Manipulability analysis for mobile manipulators". Dans *IEEE Conf. on Robotics and Automation (ICRA)*, pages 1251-1256, Séoul, 2001.
- [23] Tchon, K. et Muszynski, R. "Instantaneous kinematics and dexterity of mobile manipulators ". Dans *IEEE Conf. on Robotics and Automation (ICRA)*, volume 3, pages 2493-2498, San Francisco ,2000.

- [24] Tchon, K. et Zadarnowska, K. " Kinematic dexterity of mobile manipulators: an endogenous configuration space approach ". *Robotica*, 21 :521–530,2003.
- [25] -Bayle, B. " Modélisation et commande cinématiques des manipulateurs mobiles à roues ". Thèse de Doctorat, LAAS, Université Paul Sabatier de Toulouse ,2001.
- [26] Papadopoulos, E. et Dubowsky, S. " Coordinated manipulator/ spacedraft motion control for space robotic systems ". Dans *IEEE Conf. on Robotics and Automation (ICRA)*, pages 1696–1701 ,1991.
- [27] Luksch, T., Yl'onen, S., et Halme, A. " Combined motion control of the platform and the manipulator of Workpartner robot ". Dans *6ième Int. Conf. on Climbing and Walking Robots (CLAWAR)*, Catania, Italie, 2003.
- [28] Perrier, C., Dauchez, P., et Pierrot, F " A global approach for motion generation of non-holonomic mobile manipulators ". Dans *IEEE Conf. on Robotics and Automation (ICRA)*, pages 2971–2976,1998.
- [29] Furuno, S., Yamamoto, M., et Mohri, A. " Trajectory planning of mobile manipulator with stability considerations ". Dans *IEEE Conf. on Robotics and Automation (ICRA)*, pages 3403–3408, Taiwan,2003.
- [30] Bayle, B., Fourquet, J.-Y., et Renaud, M. " Generalized path generation for a mobile manipulator ". Dans *Int. Conf. on Mechanical Design and Production*, pages 541–545, Le Caire,2000.
- [31] Desai, J., Wang, C., Zefran, M., et Kumar, V. " Motion planning for multiple mobile manipulators ". Dans *IEEE Conf. on Robotics and Automation (ICRA)*, pages 2073–2078,1996.
- [32] Quinlan, S. et Khatib, O. " Elastic bands : connecting path planning and control ". Dans *IEEE Conf. on Robotics and Automation (ICRA)*, volume 2, pages 802–807, Atlanta, Georgia,1993.
- [33] Brock, O. et Khatib, O. " Elastic strips : real-time path modification for mobile manipulation ". Dans *Int. Symp. of Robotics Research'97*, pages 117–122,1997.
- [34] -Lamiraux, F. et Bonnafous, D. " Reactive trajectory deformation for nonholonomic systems: Application to mobile robots ". Dans *IEEE Conf. on Robotics and Automation (ICRA)*, pages 3099–3104, Washington DC ,2002.
- [35] Egerstedt, M. et Hu, X. " Coordinated trajectory following for mobile manipulation ". Dans *IEEE Conf. on Robotics and Automation (ICRA)*, pages 3479–3485, San Francisco,2000.

- [36] Canudas de Wit, C., Siciliano, B., et Bastin, G., éditeurs. Theory of robot control. Springer Verlag, 1996.
- [37] Miksch, W. et Schroeder, D. " Performance-functional based controller design for a mobile manipulator ". Dans IEEE Conf. on Robotics and Automation (ICRA), pages 227–232, Nice, 1992.
- [38] Wang, C.-C. et Kumar, V. " Velocity control of mobile manipulators ". Dans IEEE Conf. on Robotics and Automation (ICRA), volume 2, pages 713–718, 1993.
- [39] Bayle, B., Fourquet, J.-Y., Lamiroux, F., et Renaud, M. " Kinematic control of wheeled mobile manipulators ". Dans International Conference on Intelligent Robots and Systems (IROS), pages 1572–1577, 2002.
- [40] Seraji, H. " A unified approach to motion control of mobile manipulators ". International Journal of Robotics Research (IJRR), 17 :107–118, 1998.
- [41] Campion, G., d'Andrea Novel, B., et Bastin, G " Structural properties and classification of dynamic models of wheeled mobile robots " IEEE Trans. on Robotics and Automation (ITRA), 12 :47–62, 1996
- [42] - Neimark, J. et Fufaev, N. " Dynamics of Nonholonomic Systems ". Dans American Mathematical Society, volume 33, 1972.
- [43] Etienne Dombre et Wissama Khalil, "Modélisation et commande des robots", Edition Hermès Paris, 1988.
- [44] Philippe Coiffet, "La robotique: principes et applications", Edition Hermès, 1986.
- [45] John J. Craig, " Introduction to Robotics, Mechanics and Control", Addison Wesley Publishing Company, 1986.
- [46] A. Pruski, "Robotique Générale", Ellipses, 1988.
- [47] K. S. Fu R. C. Gonzalez and C. S. G. Lee, "Robotics : Control, Sensing, Vision, and Intelligence", Mc Gram-Hill Book Company, 1987.
- [48] David E. Goldberg, "Algorithmes Génétiques : Exploration, Optimisation et Apprentissage Automatique", Edition Addison Wesley France, 1994.
- [49] G. Winter, J. Periaux, M. Galan and P. Cuesta, " Genetic Algorithms in Engineering and Computer Science", British library Cataloguing in Publication Data, 1995.
- [50] G. C. and B. Chakraborty, "Ideal Marriage for Fine Tuning in GA", Proc. IEEE Int'l conf. on Systems, Man, and Cybernetics, Vol. 1, pp. 631-636, 1999.

- [52] L.A.Zadeh, " Fuzzy Sets, Information and Control" , vol.8 , pp 29-44, 1965
- [53] P. J .King and E.H. Mamdani, " The application of fuzzy control system to industrial process" , Automatica 13, 235-242, 1976.
- [54] Bernadette Bouchon Meunier, "La logique floue et ses applications", Edition Addison Wesley France, SA, 1995.
- [55] S. Dubowsky and T. D. Blubaugh, "Planning Time-Optimal Robotic Manipulator Motions and Work Places for Point-to-Point Tasks", IEEE Trans. on Robotics and Automation, Vol. 5, No. 3, pp. 377-381, June 1989.
- [56] C. J. Goh and N. Phan-Thien, "Time-Optimal Trajectories for Robot Manipulators", Robotica, Vol.9, pp. 131-138, 1991.
- [57] P. Naslin, " Théorie de la commande et conduite optimale", Dunod Paris, 1989.
- [58] James E. Bobrow, "Optimal Robot Path Planning Using the Minimum-Time Criterion", IEEE Journal of Robotics and Automation, Vol. 4, No. 4, pp. 443-454, August 1988.
- [59] Y. Uno, K. Suzuki and K. Takakura, "Trajectory Planning of Human Reaching Movements for a Time-Minimizing Task", Proc. IEEE Int'l conf. on Systems, Man, and Cybernetics, Vol. 4, pp. 1039-1044, 1999.
- [60] K. F. and H. Samet, "A Hierarchical Strategy for Path Planning Among Moving Obstacles", IEEE Trans. on Robotics and Automation, Vol. 5, No. 1, pp. 61-69, February 1989.
- [61] M. A. and C. Sanderson, "Planning Robotic Manipulation Strategies for Workpieces that Slide", IEEE Journal of Robotics and Automation, Vol. 4, No. 5, pp. 524-531, October 1988.
- [62] S. Ahmad and Shengwu Luo", Coordinated Motion of Multiple Robotic Device for Welding and Redundancy Coordination through Constrained Optimization in Cartesian Space", IEEE Trans. on Robotics and Automation, Vol. 5, No. 4, pp. 409-417, August 1989.
- [63] H. H. Tan and R. B. Potts, "A Discrete Trajectory Planner for Robotic Arms with six degrees of freedom", IEEE Trans. on Robotics and Automation, Vol. 5, No. 5, pp. 681-690, October 1989.
- [64] Stephen J. Buckley, "Teaching Compliant Motion Strategies", IEEE Trans. on Robotics and Automation, Vol. 5, No. 1, pp. 113-124, February 1989.
- [65] Z. Li and John Canny, "Motion of Two Rigid Bodies with Rolling Constraint", IEEE Trans. on Robotics and Automation, Vol. 6, No. 1, pp. 62-72, February 1990.

- [66] Yaping Dai, "Use of Pseudo Measurement to Real-Time Target Tracking", Proc. IEEE Int'l conf. on Systems, Man, and Cybernetics, Vol. 5, pp. 33-38, 1999.
- [67] J.A.,A.Rojas and C.S.Cajun,"Trajectory Planning in Robotics Continuous-Path Applications", IEEE Journal of Robotics and Automation, Vol.4, No.4, pp.380-385, August 1988.
- [68] J. Feddema and O. R. Mitchell, "Vision-Guided Servoing with Feature-Based Trajectory Generation", IEEE Trans. on Robotics and Automation, Vol. 5, No. 5, pp. 691-700, October 1989.
- [69] Simon X. Yang and Max Meng, "Real-time Collision-free Path Planning of Robot Manipulators using Neural Network Approaches", Autonomous Robots, Vol.9, No. 1, pp. 27-33, 2000.
- [70] Sun-Yuan Kung and J. N. Hwang," Neural Network Architectures for Robotic Applications," IEEE Trans. on Robotics and Automation, Vol. 5, No. 5, pp. 641-657, October 1989.
- [71] H. M. Tai, J. Wang and K. Ashenayi, "A neural Network-Based Tracking Control System", IEEE Trans. on Robotics and Automation, Vol. 39, No. 6, pp. 504-510, December 1992.
- [72] P. Payeur, H. Le-Huy and C. M Gossolin, "Trajectory Prediction for Moving Objects Using Artificial Neural Networks", IEEE Trans. on Industrial Electronics, Vol. 42, No. 2, pp. 147-158, April 1995.
- [73] X. Yang and Max Meng, "A neural Network Approach to Real-Time Motion Planning and Control of Robot Manipulators", Proc. IEEE Int'l conf. on Systems, Man, and Cybernetics, Vol. 4, pp. 674-679, October 1999.
- [74] T. N., K. Sugawara and I. Yoshihara, "A Motion Planning Method for a Hyper Multi-joint Manipulator using Genetic Algorithm", Proc. IEEE Int'l conf. on Systems, Man, and Cybernetics, V. 4, pp. 645-650, 1999.
- [75] K. T. Hisashi Tamaki, "A Continuous Age Model of Genetic Algorithms Applicable to Optimization Problems with Uncertainties", Proc. IEEE Int'l conf. on Systems, Man, and Cybernetics, Vol. 1, pp. 637-642, 1999.
- [76] Q. C. Meng and Z. Chen, "Genetic Algorithms Encoding Study and A sufficient Convergence Condition of GAs", Proc. IEEE Int'l conf. on Systems, Man, and Cybernetics, Vol. 1, pp. 649-652, 1999.
- [77] Rolf Johansson, "Adaptive Control of Robot Manipulator Motion", IEEE Trans. on Robotics and Automation, Vol. 6, No. 4, pp. 483-490, August 1990.

- [78] K. S., Y. I. and M. Ohtani, "Optimal State Feedback Control Law for Nonlinear Systems and its Best Approximation by a Neural Network", Proc. IEEE Int'l conf. on Systems, Man, and Cybernetics, Vol. 4, pp. 1096-1101, 1999.
- [79] "W. J. Wang, S. S. Lu and Chen-Fa Hsu, "Experiments on the Position Control of a One-Link Flexible Robot Arm", IEEE Trans. on Robotics and Automation, Vol. 5, No. 3, pp. 373-377, June 1989.
- [80] Lin Guo and J. Angeles, "Controller Estimation for the Adaptive Control of Robotic Manipulators", IEEE Trans. on Robotics and Automation, Vol. 5, No. 3, pp. 315-323, June 1989.
- [81] S. R. Oh, Z. Bien and IL Hong Suh," An Iterative Learning Control Method with Application for the Robot Manipulator", IEEE Journal of Robotics and Automation, Vol. 4, No. 5, pp. 508-514, October 1988.
- [82] M. H. Liu, W. S. Chang and L. Q. Zhang," Multivariable Self-Tuning Control of Redundant Manipulators", IEEE Journal of Robotics and Automation, Vol. 4, No. 5, pp. 498-507, October 1988.
- [83] M. W. Walker, " Adaptive Control of Manipulators Containing Closed Kinematic Loops", IEEE Trans. on Robotics and Automation, Vol. 6, No. 1, pp. 10-19, February 1990.
- [84] John S. Reed and Petros A. Ioannou, "Instability Analysis and Robust Adaptive Control of Robotic Manipulators", IEEE Trans. on Robotics and Automation, Vol. 5, No. 3, pp. 381-386, June 1989.
- [85] J. Murray and G. H. Lovell, "Dynamic Modeling of Closed-chain Robotic Manipulators and Implications for Trajectory Control", IEEE Trans. on Robotics and Automation, Vol. 5, No. 4, pp. 522-528, August 1989.
- [86] E. B. Braiek and H. Belkhiria, "Robustness Stability Measure for a Class on Nonlinear Uncertain Systems", Proc. IEEE Int'l conf. on Systems, Man, and Cybernetics, Vol. 1, pp. 28-32, 1999.
- [87] H. Seraji, "Configuration Control of Redundant Manipulators: Theory and Implementation", IEEE Trans. on Robotics and Automation, Vol. 5, No. 4, pp. 472-490, August 1989.
- [88] J. Lai an D. Yang, "An Efficient Motion Control Algorithm for Robots with Wrist Singularities", IEEE Trans. on Robotics and Automation, Vol. 6, No. 1, pp. 113-117, February 1990.

- [89] Y. K., S. Omatu and M. Yoshioka, "Neuro-PID Control for Stabilization of Inverted Single and Double Pendulums", Proc. IEEE Int'l conf. on Systems, Man, and Cybernetics, Vol. 4, pp. 1108-1112, 1999.
- [90] M. Asano, T. Yamamoto, T. Oki and M. Kaneda, "A Design of Neural-Net Based Predictive PID Controllers", Proc. IEEE Int'l conf. on Systems, Man, and Cybernetics, Vol. 4, pp. 1113-1118, 1999.
- [91] Y. G. Leu, W. Y. Wang, and T. T. Lee, " H_{∞} -Observer-Based Adaptive Fuzzy- Neural Control for a Class of Uncertain Nonlinear Systems", Proc. IEEE Int'l conf. on Systems, Man, and Cybernetics, Vol. 1, pp. 449-454, 1999.
- [92] W. Chang, R. Hwang and J. Hsieh, " A Single Auto-Tuning Neural Controller for Nonlinear Dynamic Systems", Proc. IEEE Int'l conf. on Systems, Man, and Cybernetics, Vol. 1, pp. 461-465, 1999.
- [93] M. Saito and J. Sugi, "A Hierarchical Bayesian Scheme for Nonlinear Dynamical System Reconstruction and Prediction with Neural Nets", Proc. IEEE Int'l conf. on Systems, Man, and Cybernetics, Vol. 4, pp. 1119-1124, 1999.
- [94] C. M. Liaw and S. Y. Cheng, "Fuzzy Two-Degrees-of-Freedom Speed Controller for Motor Drives", IEEE Trans. on Industrial Electronics, Vol. 42, No.2, pp. 209-216, April 1995.
- [95] G. Sousa, B. Bose and John G. Cleland, "Fuzzy Logic Based On-line Efficiency Optimization Control of an Indirect Vector-Controlled Induction Motor Drive", IEEE Trans. on Industrial Electronics, Vol. 42, No.2, pp. 192-198, April 1995.
- [96] Y. Hanaki, T. Hashiyama and S. Okuma, "Accelerated Evolutionary Computation Using Fitness Estimation", Proc. IEEE Int'l conf. on Systems, Man, and Cybernetics, Vol. 1, pp. 643-648, 1999.
- [97] Paul V. Goode and Mo-yuen chow, "Using a Neural/Fuzzy System to Extract Heuristic knowledge of Incipient Faults in Induction Motors", IEEE Trans. on Industrial Electronics, Vol. 42, No.2, pp. 131-146, April 1995.
- [98] C. L. Hwang, Y. H. Chen and C. Hung Lin, "A discrete-Time Multivariable Neuro-Adaptive Control for Nonlinear Unknown Dynamic Systems", Proc. IEEE Int'l conf. on Systems, Man, and Cybernetics, Vol. 4, pp. 1102-1107, 1999.

- [99] W. T. Miller, R. P. Hewes and L. G. Kraft, "Real-Time Dynamic Control of Industrial Manipulator Using a Neural-Network-Based Learning Controller", IEEE Trans. on Robotics and Automation, Vol. 6, No. 1, pp. 1-8, February 1990.
- [100] K. S. Narendra and K. P., "Identification and Control of Dynamical Systems Using Neural Networks", IEEE Trans. on Neural Networks, Vol. 1, No. 1, pp. 4-26, March 1990.
- [101] Lo and Kuo, "Decoupled Fuzzy Sliding-Mode Control", IEEE Trans. on Fuzzy Systems, Vol. 6, No. 3, pp. 217-222, August 1998.
- [102] C. Kung and Ti-Hung Chen, "Fuzzy Sliding Mode Controller Design Based on Fuzzy Union Reasoning", Proc. IEEE Int'l conf. on Systems, Man, and Cybernetics, Vol. 3, pp. 320-325, 1999.
- [103] H. Shih Lee, "An Optimal Aggregation Method for Fuzzy Opinions of Group Decision", Proc. IEEE Int'l conf. on Systems, Man, and Cybernetics, Vol. 3, pp. 314-319, 1999.
- [104] Eric Tsang and D. S. Yeung, " Optimizing Fuzzy Knowledge Base by Genetic Algorithms and Neural Networks", Proc. IEEE Int'l conf. on Systems, Man, and Cybernetics, Vol. 3, pp. 367-371, 1999.
- [105] Y. T. and Y. Dote, "Novel Fuzzy-Neural Network with General Parameter Learning Applied to Sliding Mode Control Systems, Proc. IEEE Int'l conf. on Systems, Man, and Cybernetics, Vol. 1, pp. 376-379, 1999.
- [106] T. Morimoto, Kang Tu and Y. Hachimoto, "A Decision and Control Technique Based on Fuzzy Control, Neural Networks, and Genetic Algorithms For Optimization of a Fruit-Storage Process", Proc. IEEE Int'l conf. on Systems, Man, and Cybernetics, Vol. 6, pp. 440-445, 1999.
- [107] Arit Thammano, "A New Forecasting Approach with Neuro-Fuzzy Architecture", Proc. IEEE Int'l conf. on Systems, Man, and Cybernetics, Vol. 1, pp. 386-389, 1999.
- [108] K. Kiguchi, Hyeon-Ho Jang and Toshio Fukuda, "Identification of Robot Manipulators Using Neural Networks and Genetic Programming", Proc. IEEE Int'l conf. on Systems, Man, and Cybernetics, Vol. 4, pp. 802-806, 1999.
- [109] F. Abdessamed " Application de la logique floue à la commande de processus complexes de la manipulation mobile " Thèse de Doctorat, Laboratoire de Robotique de Paris ,2003.
- [110] M. Suzuki, N. Mizuno, Y. Y. and K. Matsunami, "Reformulation of Dynamics Equation of Planar Two-Joint Arm Movements", Proc. IEEE Int'l conf. on Systems, Man, and Cybernetics, Vol. 4, pp. 807-812, 1999.

- [111] Jindong T., Ning Xi and Yuechao Wang, "Integrated Task Planning and Control for Mobile Manipulators", The International Journal of Robotics Research, vol. 22, no. 5, pp. 337-354, May 2003.
- [112] M .Hamani, A.Hassam and A.Khellaf. "Motion Planning and Control of Robot" SNCS'2001 Séminaire National sur le contrôle et les signaux , octobre 2001
- [113] M .Hamani, A.Hassam and A.Khellaf. Motion Planning and Control of Robot Manipulators. ISC2001 , IASTED International Conference on intelligent Systems and control CANADA November 19 to 22, 2001.
- [114] A.Hassam, K.Benmahammed and M..Hamani "Motion Planning and Control of Mobile Manipulators" Asian Journal of Information Technology, 5 (9): 938-943, 2006.
- [115] Yanrong Hu and Simon X. Yang, "A Knowledge Based Genetic Algorithm for Path Planning of *a* Mobile Robot", Proc. IEEE Int'l. Conf. on robotics and automation, pp. 4350-4355, April 2004.
- [116] T. Nishimura, K. Sugawara, I. Yoshihara, and K. Abe, "A Motion Planning Method for *a* Hyper Multi-joint Manipulator using Genetic Algorithm", Proc. IEEE Int'l. Conf. on Systems, Man, and Cybernetics, vol. 4, pp. 645-650, October 1999.
- [117] Bruno Siciliano and Luigi Villani, "Two-Time Scale Force and Position Control of Flexible Manipulator", IEEE Int'l. Conf. on robotics and automation, pp. 2729-2734, May 2001.
- [118] Seok Jo Go, Min Cheol Lee and Min Kyu Park, " Fuzzy-Sliding Mode of Automatic Polishing Robot System with the Self Tuning Fuzzy Inference Based on Genetic Algorithm", Proc. IEEE Int'l. Conf. on robotics and automation, pp. 2962-2967, May 2001.
- [119] M.Fruchard" Methodologies pour la commande de manipulateurs mobiles non-holonomes" Thèse de Doctorat, Ecole Nationale Supérieure des mines, Paris ,2005.