



الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieure et de la Recherche Scientifique

جامعة وهران 2 محمد بن أحمد
Université d'Oran 2 Mohamed Ben Ahmed
معهد الصيانة والأمن الصناعي
Institut de Maintenance ET de Sécurité Industrielle

Département: Maintenance en Instrumentation

MÉMOIRE

En vue de l'obtention du diplôme de Master

Filière : Génie Industriel

Spécialité : Ingénierie de la maintenance en Instrumentation

Thème

***Contribution à L'optimisation De La Trajectoire
D'un Robot à L'aide De La Méthode PSO***

Présenté et soutenu publiquement par :

MENDAS ZOBIDA

Devant le jury composé de:

Nom et Prénom	Grade	Etablissement	Qualité
BOUHAFS Mohamed	MCA	IMSI-Univ D'Oran2	Président
LALAOUI Mohamed El Amine	MAA	IMSI-Univ D'Oran2	Encadreur
KACIMI Abderrahmane	MCB	IMSI-Univ D'Oran2	Examineur

Année universitaire : 2022/2023.



Remerciement

Nous remercions tout d'abord, Allah qui nous a donné la force et le courage pour terminer nos études et élaborer ce modeste travail.

J'exprime mon sincère remerciement à toutes les personnes qui ont aidés de près ou de loin pour réaliser ce travail.

Ces remerciements sont adressés chaleureusement à notre encadreur Mr Lalaoui Mohamed El Amine pour nous avoir guidé et encouragé durant ce travail.

Nous tenons à remercier aussi l'ensemble des enseignants et responsables du département maintenance en instrumentation qui ont contribué à amener à bien notre formation par leurs aides et leurs conseils.

Grand Merci à tous !





Dédicace

À Ma mère qui m'a soutenu durant tout mon parcours scolaire et professionnel, pour tous ses sacrifices, son amour, sa tendresse, son soutien et ses prières tout au long de mes études.

J'espère un jour te rendre ce que tu m'as donné, je t'aime maman.

À Mon père qui m'a donné le support et l'encouragement nécessaire afin de concrétiser mes ambitions.

À mes chères frères et sœurs, qui ont partagé avec moi tous les moments que ce soient durs ou adorables, et qui me souhaitent toujours de réussir.

À mes adorables neveux : mes princesses Wessal et Razan.

À mes amis, à tous ceux que j'aime, et tous ceux

Qui me sont chers, je dédie ce modeste travail.

Zobida.



Résumé

Ce projet de fin d'études propose une approche novatrice pour l'optimisation des trajectoires de robots en utilisant la méthode de l'optimisation par essaim de particules (PSO - Particle Swarm Optimization). L'optimisation des trajectoires est une composante cruciale de la robotique, avec des applications diverses allant de la navigation autonome des robots mobiles à la planification de trajectoires pour des bras robotiques industriels.

La méthode PSO est un algorithme d'optimisation inspiré du comportement collectif des essaims d'oiseaux. Elle présente une grande efficacité pour résoudre des problèmes d'optimisation complexes. Dans ce projet, nous avons adapté la méthode PSO pour résoudre le défi de l'optimisation des trajectoires de robots. L'objectif principal est de permettre à un robot de déterminer la trajectoire la plus optimale pour atteindre sa destination tout en évitant les obstacles et en prenant en compte des contraintes diverses telles que la consommation d'énergie ou le temps de parcours.

L'approche développée dans ce projet a été testée avec succès sur des scénarios de simulation et a démontré des améliorations significatives par rapport aux méthodes d'optimisation traditionnelles. Les résultats obtenus montrent que la méthode PSO peut être une solution prometteuse pour l'optimisation des trajectoires robotiques dans des environnements complexes et dynamiques.

Ce travail de recherche contribue ainsi à l'avancement de la robotique en fournissant une méthode innovante et efficace pour améliorer la planification de trajectoires des robots, ouvrant la voie à des applications plus sûres et plus performantes dans divers domaines industriels et technologiques.

Les mots clefs :

Planification de trajectoires, Algorithme d'optimisation, Intelligence artificielle, Méthodes d'optimisation avancées, Dynamique des robots, PSO (Particle Swarm Optimization).

Abstract

This final year project proposes an innovative approach for optimizing robot trajectories using the Particle Swarm Optimization (PSO) method. Trajectory optimization is a critical component of robotics, with diverse applications ranging from autonomous navigation of mobile robots to trajectory planning for industrial robotic arms.

PSO is an optimization algorithm inspired by the collective behavior of bird flocks. It demonstrates high efficiency in solving complex optimization problems. In this project, we have adapted the PSO method to address the challenge of optimizing robot trajectories. The main objective is to enable a robot to determine the most optimal trajectory to reach its destination while avoiding obstacles and considering various constraints such as energy consumption and travel time.

The approach developed in this project has been successfully tested in simulation scenarios and has shown significant improvements compared to traditional optimization methods. The results obtained indicate that the PSO method can be a promising solution for optimizing robot trajectories in complex and dynamic environments.

This research work contributes to the advancement of robotics by providing an innovative and effective method to enhance robot trajectory planning, paving the way for safer and more efficient applications in various industrial and technological domains.

Key words:

Trajectory Planning, Optimization Algorithm, Artificial Intelligence, Advanced Optimization Methods, Robot Dynamics, Particle Swarm Optimization (PSO).

LISTE DES FIGURES

Figures	Page
Figure.I.1: le canard digérateur.	5
Figure.I.2. De gauche à droite : (a) horloge à l'eau de Clepsydre , (b) Théâtre de l'automate du Héros d'Alexandrie, (c) Machines hydrauliques hydro-alimentées humanoïdes d'Al-Jazari [1].	5
Figure I.3. Structure fonctionnelle d'un robot	6
Figure.I.4. Robots mobiles.	7
Figure.I.5. Robot manipulateur	7
Figure.I.6. Bras manipulateur parallèle.	7
Figure I.7 : Naive bayes	11
Figure I.8 : Support Vector Machine	12
Figure I.9 : Illustration du cadre général de l'apprentissage par renforcement.	13
Figure.I.10: Neurone biologique	17
Figure.I.11 : Neurone artificiel	18
Figure.I.12: Fonction linéaire	20
Figure.I.13: Fonction sigmoïde	20
Figure.I.14 : Réseaux de neurone bouclés.	21
Figure.I.15 : Réseaux de neurone non bouclés.	22
Figure.I.16 : Figuration de l'apprentissage supervisé	22
Figure.I.17: Figuration de l'apprentissage non supervisé.	23
Figure.I.18: Modèle d'un perceptron simple.	25
Figure.I.19 : Modèle de perceptron multicouches.	25
Figure.I.20: Formes usuelles des fonctions d'appartenance.	28
Figure II.1 : La structure d'un robot.	34
Figure II.2 : La représentation schématique de la composition d'un robot.	35
Figure II.3.a : Représentation d'une Articulation rotoïde	36
Figure II.3.b: Représentation d'une Articulation prismatique	36
Figure II.4 : Structure d'un manipulateur d'architecture sérielle	37
Figure II.5 : Représentation d'une chaîne cinématique fermée	37
Figure II.6 : Machine parallèle de type Delta	38
Figure II.7: Robot cartésien.	39

Figure.II.8 : Structure d'un bras manipulateur sphérique.	39
Figure II.9: Robot cylindrique.	40
Figure II. 10 Robot SCARA.	40
Figure II.11: Robot 3R.	41
Figure II.12 : Transformation des repères.	43
Figure II.13 : Transformation de translation pure	43
Figure II.14 : Robot à structure ouverte simple	45
Figure II.15: Paramètres de Denavit-Hartenberg	46
Figure II.16 : Schéma général de la commande adaptative	54
Figure III.1: Diagramme du bras pelican à deux degré de liberté (2ddl).	61
Figure III.2 : Planification dans l'espace articulaire.	70
Figure III.3 : Planification et génération de trajectoires.	71
Figure. III.4 : Représentation de l'opération de mutation	81
Figure IV.1 : Trajectoire point par point.	91
Figure IV.2 : Robot planaire 2R 2ddl.	95
Figure IV.3 : Trajectoire du robot dans l'espace cartésien.	96
Figure IV.4 : Valeur optimale de la fonction objective.	96
Figure IV.5 : Trajectoires de positions angulaires articulaires.	97
Figure IV.6 : Trajectoires de vitesses angulaires articulaires.	97
Figure IV.7 : Trajectoires d'accélération angulaires articulaires.	98
Figure IV.8 : Couples articulaires.	98
Figure IV.9 : Temps consommé par les trajectoires d'un point à un autre.	99
Figure IV.10 : Distance totale parcourue par une articulation.	99
Figure IV.11 : Longueur totale cartésienne de la trajectoire.	100
Figure IV.12 : Couple total excessif.	100

LISTE DES TABLEAUX

N°	Tableaux	Page
01	Tableau I.1 : Analogie entre le neurone biologique et le neurone artificiel.	19
02	Tableau.I.2 : La loi de Hebb.	24
03	Tableau I.3 : Comparaison entre la logique floue et les réseaux de neurones	29
04	Tableau II.1 – Comparaison des manipulateurs d’architectures sérielle et parallèle	38
05	Tableau III.1: paramètres du robot (pelican).	62

Liste des abréviations

RM : Robot manipulateur

PSO : Optimisation par Essaim Particulaire (particle swarm optimization)

IA : L'intelligence Artificielle

ML : Machine Learning (apprentissage automatique)

SL: Supervised learning (apprentissage supervisé)

SVM : machine à vecteurs de support

RL : Reinforcement Learning (apprentissage par Renforcement)

MDP : Processus de décision markovien

SARA: (State-Action-Reward-State-Action)

DQN: Deep Q-Network

DL: Deep Learning (apprentissage profond)

ANN: Artificial Neural Network (Réseau neurones Artificiel)

RNN: Recurrent Neural Network (Réseau neurones récurrent)

CNN: Convolutional Neural Networks (Réseau neurones convolutifs)

MPL: Multi-Layer Perceptron (perceptron multicouche)

LSTM: Long Short-Term Memory

MCD : Modèle cinématique direct

MCI : Modèle cinématique inverse

MD : Modèle dynamique

MDD : Modèle dynamique direct

MDI : Modèle dynamique inverse

MGD : Modèle géométrique direct

MGI : Modèle géométrique inverse

LE : Lagrange-Euler

OT: Organe terminal (effecteur)

D-H: Denavite-Hartenberg

Ddl : Degrés de liberté

P : Coordonnées homogènes

T : Transformation homogène

X : Les coordonnées opérationnelles

IP : indice de performance

Γ : vecteur des forces généralisées (ou les couples) appliquées au système.

K : l'énergie cinétique.

P : l'énergie potentiel.

q : vecteur des variables articulaires généralisées.

\dot{q} : vecteur des vitesses angulaires.

\ddot{q} : vecteur des accélérations angulaires.

M : matrice d'inertie.

C : matrice spécifiant l'effet de centrifuge et de Coriolis.

g : vecteur exprimant l'effet gravitationnel.

τ : vecteur des couples généralisés.

x_k : est le vecteur de paramètres à l'itération n

α : est le taux d'apprentissage (ou pas d'apprentissage),

$\nabla f(x_k)$: est le gradient de la fonction objectif f évalué au point x_k .

GP : La Programmation Génétique (Genetic Programming)

P_{best}^i : best meilleure trajectoire trouvée par la particule i .

P_{best}^g : best meilleure trajectoire trouvée par toutes les particules.

SOMMAIRE

Remerciement	I
Dédicace	II
Résumé	III
Liste des figures	V
Liste des tableaux	VII
Liste des abréviations	VIII
Introduction générale	1
Chapitre I : état de l'art sur les approches de contrôle en robotique par les techniques de l'IA	
I. Introduction :.....	4
I.1 Historique :.....	4
I.1.1 Les automates:	4
I.1.2 Intelligence artificielle:.....	5
I.1.3 Histoire de la robotique :	5
I.2 La robotique :	6
I.2.1 Définition :	6
I.2.2 Type des robots :.....	6
I.2.2.1 Robots mobiles :.....	6
I.2.2.2 Robots manipulateurs :.....	7
I.2.3 Domaine d'utilisation des robots :.....	7
I.3 Les techniques classiques de contrôle en robotique :.....	8
I.4 L'intelligence Artificielle En Robotique :	9
I.5 Les techniques d'intelligence artificielle en robotique :.....	10
I.5.1 L'apprentissage automatique(ML):	10
I.5.1.1 Méthodes d'apprentissages automatiques :.....	10
I.5.1.1.1 L'apprentissage supervisé (Supervised Learning)	10
I.5.1.1.2 L'apprentissage non supervisé (Unsupervised Learning)	11
I.5.2 L'apprentissage par renforcement (RL) :	12
I.5.2.1 Les algorithmes d'apprentissage par renforcement :	14
I.5.2.2 Avantages Et Inconvénients De l'apprentissage par renforcement :.....	15
I.5.3 L'apprentissage profond (Deep Learning) :	15

I.5.4	Les réseaux de neurone artificiels:	16
I.5.4.1	Généralité sur les réseaux de neurones :	16
I.5.4.1.1	Neurones biologique.....	17
I.5.4.1.2	Neurones Formel(Artificiel).....	17
I.5.4.2	Architecture des réseaux de neurones :	20
I.5.4.2.1	Les réseaux de neurones bouclés.....	21
I.5.4.2.2	Les réseaux de neurones non bouclés.....	21
I.5.4.3	L'apprentissage d'un réseau de neurones :	22
I.5.4.4	Règles d'apprentissage des réseaux de neurones :	23
I.5.4.5	Différents modèles des réseaux de neurones :	24
I.5.5	La logique floue :	26
I.5.5.1	Principe de la logique floue :	26
I.5.5.2	Notions de base de la logique floue :	27
I.5.5.3	Les types de systèmes flous:	28
I.5.5.4	L'utilisation de la logique floue :	29
I.5.5.5	Avantages Et Inconvénients De La Logique Floue Et Des Réseaux De Neurones :	29
I.6	Problèmes et défis de l'utilisation de l'IA dans le contrôle robotique :	30
I.7	Conclusion :	31

Chapitre II : état de l'art sur les commandes et les modélisations appliquées aux robots manipulateur

II.1	Introduction	33
II.2	Notion générales sur les robots manipulateurs.....	33
II.2.1	Structure et composants technologiques d'un RM :	34
II.2.2	Articulations et degrés de liberté	36
II.2.3	Manipulateurs d'architecture sérielle et d'architecture parallèle.....	36
II.2.3.1	Les manipulateurs d'architecture sérielle :	37
II.2.3.2	Les manipulateurs d'architecture parallèle :	37
II.2.4	Classification géométrique des robots manipulateurs	39
II.3	Modélisation des robots manipulateurs	41
II.3.1	Modélisation géométrique :	41
II.3.1.1	Transformation homogène et coordonnées généralisées :	41
II.3.1.2	Modèle géométrique direct:.....	44
II.3.1.2.1	Paramètres de Denavit-Hartenberg.....	44

II.3.1.3	Modèle géométrique inverse :.....	47
II.3.2	Modélisation Cinématique :.....	48
II.3.2.1	Modèle Cinématique direct :.....	48
II.3.2.2	Modèle Cinématique inverse :.....	50
II.3.3	Modélisation dynamique :.....	51
II.3.3.1	Modèle dynamique direct (MDD) :.....	51
II.3.3.2	Modèle dynamique inverse (MDI) :.....	51
II.3.3.3	Formalisme de Newton-Euler :.....	51
II.3.3.4	Formalisme de Lagrange :.....	52
II.4	Commande Des Robots Manipulateurs.....	53
II.4.1	Commande prédictive:.....	53
II.4.2	Commande adaptative :.....	54
II.4.2.1	Limitations de la Commande Adaptative :.....	55
II.4.3	Commande par articulation :.....	55
II.4.4	Commande jacobienne :.....	55
II.4.5	Commande passive :.....	56
II.4.6	Commande optimale :.....	57
II.4.7	Commande robuste :.....	57
II.5	Conclusion :.....	58

Chapitre III : recherche sur l'optimisation de la trajectoire d'un robot manipulateur à 2 DDI

III.1	Introduction.....	60
III.2	Modélisation du robot manipulateur à deux ddi.....	60
III.2.1	Description du robot manipulateur :.....	60
2.1.1	Domaine d'application :.....	61
III.2.2	Modélisation dynamique :.....	62
III.3	Notion générale sur l'optimisation de trajectoire.....	67
III.3.1	Les définitions de la trajectoire et de l'optimisation :.....	67
III.3.1.1	Chemins et trajectoires :.....	67
III.3.2	Planification des chemins et planification des trajectoires :.....	69
III.3.2.1	Planification des chemins :.....	69
III.3.2.2	Planification des trajectoires:.....	69
III.3.2.3	Génération de la trajectoire dans l'espace articulaire :.....	70
III.3.3	Génération de trajectoire en environnement connu :.....	71
III.3.4	Critères d'optimisation pour la planification des trajectoires.....	72

III.3.4.1	Minimisation du temps :	72
III.3.4.2	Minimisation de l'énergie :	72
III.3.4.3	Minimisation du jerk :	72
III.3.5	Représentation mathématique de la trajectoire :	73
III.4	Méthodes d'Optimisation de Trajectoire	74
III.4.1	Méthodes Classiques :	74
III.4.1.1	Méthode de Newton-Raphson :	74
III.4.1.2	La Méthode de descente de gradient :	77
III.4.2	Algorithmes d'Optimisation Heuristiques:	78
III.4.2.1	Les algorithmes génétiques:	79
III.4.2.2	Optimisation par Essaim Particulaire (PSO).....	82
III.4.2.3	Recuit simulé.....	84
III.4.2.4	Recherche tabou.....	86
III.4.2.4.1	Principes de la recherche tabou.....	86
III.4.2.4.2	Planification de trajectoire à l'aide de la recherche taboue.....	87
III.5	Conclusion :	89

Chapitre IV : recherche sur les algorithmes PSO (particule, swarm optimization), simulation et interprétation du résultat

IV.1.	Introduction	Erreur ! Signet non défini.
IV. 2.	Planification de trajectoires par l'algorithme essaim des particules	92
IV. 3.	Paramètres de l'algorithme PSO	93
IV. 4.	Critère de la fonction objective	94
IV. 5.	Résultats des simulations	96
IV. 6.	Conclusion	106
	Conclusion générale.....	108
	Bibliographie.....	

Introduction générale

Introduction générale

La robotique, en constante évolution, a transformé la façon dont nous abordons un large éventail de domaines, de l'industrie à la médecine en passant par l'exploration spatiale. Au cœur de cette révolution technologique se trouve la capacité des robots à se déplacer et à interagir avec leur environnement. La planification de trajectoires, un aspect essentiel de la robotique, détermine la manière dont un robot se déplace dans un espace donné pour accomplir une tâche spécifique. Que ce soit pour la navigation autonome de véhicules terrestres, la manipulation d'objets par des bras robotiques industriels, ou l'exploration de terrains inconnus par des robots spatiaux, la qualité de la planification de trajectoires a un impact direct sur la performance, la sécurité et l'efficacité des robots.

L'optimisation des trajectoires robotiques est une problématique complexe et multidimensionnelle. Elle exige la prise en compte de multiples facteurs tels que :

- La distance à parcourir,
- Les obstacles à éviter,
- La consommation d'énergie,
- Le temps de parcours,

Et bien d'autres contraintes spécifiques à chaque application.

Dans cette perspective, l'optimisation des trajectoires se révèle être un défi de taille, sollicitant des solutions innovantes et des outils avancés.

Ce projet de fin d'études se concentre précisément sur l'amélioration de la planification de trajectoires des robots en ayant recours à la méthode de l'optimisation par essaim de particules (PSO - Particle Swarm Optimization). Cette méthode, inspirée du comportement collectif d'essaims d'oiseaux, a démontré sa capacité à résoudre des problèmes d'optimisation complexes dans divers domaines. L'objectif principal de cette recherche est d'adapter la méthode PSO pour résoudre le défi de l'optimisation des trajectoires robotiques, en permettant aux robots de déterminer des trajectoires optimales tout en respectant les contraintes imposées par l'environnement et les objectifs de la mission.

Au cours de cette étude, nous explorerons en détail la méthodologie et les résultats obtenus, mettant en évidence les avantages et les améliorations significatives apportés par l'approche PSO par rapport aux méthodes d'optimisation traditionnelles. Nous examinerons également les implications de cette recherche sur l'avancement de la robotique, en mettant en lumière les

applications potentielles dans des domaines variés, allant de l'industrie à l'espace en passant par la médecine.

Cette contribution à l'optimisation des trajectoires robotiques représente une étape importante vers la réalisation de robots plus performants, plus intelligents et plus adaptables, ouvrant la voie à un avenir où la robotique joue un rôle central dans la résolution de défis complexes de Notre société moderne.

Dans le cadre de notre projet de fin d'études, nous nous intéressons à la Contribution à L'optimisation De La Trajectoire D'un Robot à L'aide De La Méthode PSO.

Le présent mémoire est structuré en 4 chapitres :

Chapitre 1 : : état de l'art sur les approches de contrôle en robotique par les techniques d'IA

Chapitre 2 : état de l'art sur les commandes et les modélisations appliquées aux robots manipulateurs

Chapitre 3 : recherche sur l'optimisation de la trajectoire d'un robot manipulateur à 2 ddl

Chapitre 4 : recherche sur les algorithmes PSO (**Particle Swarm Optimization**), et simulation, interprétation du résultat.

Nous concluons notre mémoire par une conclusion générale et une bibliographie.

Chapitre I

État de l'art sur les approches de contrôle en robotique par les techniques de l'IA

I. Introduction :

La robotique est considérée comme un des grands axes de progrès scientifiques et techniques, elle consiste à l'automatisation de nombreux secteurs de l'activité humaine afin par exemple, d'augmenter la productivité dans les entreprises, stimuler la compétitivité entre eux et surtout pouvoir travailler dans des zones à haut risque (espace, démantèlement nucléaire, déminage ...etc.).

Le secteur d'activité des systèmes robotiques s'est élargi de façon importante en Particulier dans le domaine médical, spatial, industriel, militaire et dans le domaine d'agriculture.

La robotique est un domaine en constante évolution qui vise à concevoir et développer des machines capables d'accomplir des tâches de manière autonome.

Dans ce contexte, le contrôle des robots joue un rôle crucial pour assurer leur bon fonctionnement et leur capacité à interagir efficacement avec leur environnement. Au fil des années, les techniques d'intelligence artificielle (IA) ont connu des avancées significatives, offrant de nouvelles possibilités dans le domaine du contrôle robotique.

Ce chapitre vise à dresser un état de l'art des approches de contrôle en robotique utilisant les techniques d'IA, en mettant l'accent sur les principaux paradigmes et les méthodes les plus couramment utilisées.

I.1 Historique :

Vers 3500 avant J-C la roue est apparue et considérée comme le premier ouvrage technique significatif de l'histoire de l'humanité. Les automates ont reposé sur des ressort, engrenage et autre mécanisme, et puis l'informatique est venu changer la donne en permettant de stocker de très large quantité d'information, et de séquences d'action dans une petite puce. Cela a entraîné une sophistication du robot et a permis de doter le robot d'une intelligence artificiel. [2]

I.1.1 Les automates:

Le XVIIIe siècle est l'âge d'or des automates, et l'un des automates les plus étonnants est sans doute le Canard à digérer de Jacques de Vaucanson* (1738). Capable de boire, de manger, de bavarder et de digérer comme un véritable animal, il a malheureusement été détruit dans un incendie au milieu des années 1800 et il ne reste que des photographies.

Chapitre I : État de l'art sur les approches de contrôle en robotique par les techniques de l'IA

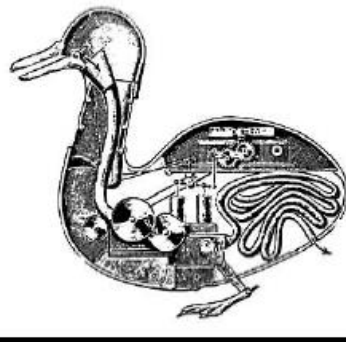


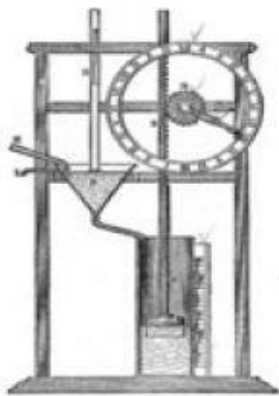
Figure.I.1: le canard digérateur [3].

I.1.2 Intelligence artificielle:

Selon la cybernétique, l'objectif de rendre les machines intelligentes a conduit Alan Turing en 1950 à définir une méthode pour tester la présence ou l'absence de conscience ou d'intelligence au sein des machines.

I.1.3 Histoire de la robotique :

Depuis la nuit des temps, un des rêves essentiels de l'homme, c'est de créer une machine intelligente à son image. Ce rêve fait désormais partie de la réalité frappante de notre monde. Le concept du robot a été établi par de nombreuses réalisations historiques créatives, tel que : l'horloge à l'eau de Clepsydre introduite par les Babyloniens (1400 avant JC), le théâtre de l'automate du Héros d'Alexandrie (100 après JC), les machines hydrauliques hydroalimentées et les humanoïdes d'Al-Jazari (1200), fig.I.1 et les nombreux designs ingénieux de Léonard de Vinci (1500). Néanmoins, l'émergence du robot physique a dû attendre l'avènement de ses technologies sous-jacentes au cours du XXe siècle [1].



(a)



(b)



(c)

Figure.I.2. De gauche à droite : (a) horloge à l'eau de Clepsydre, (b) Théâtre de L'automate du Héros d'Alexandrie, (c) Machines hydrauliques hydro-alimentées humanoïdes d'Al-Jazari [1].

I.2 La robotique :

I.2.1 Définition :

La robotique est un ensemble de méthodes et de techniques (mécanique, électronique, automatique, informatique) utilisées pour la conception et la programmation de robots pour faciliter la vie humaine.

➤ **Le robot** : est un dispositif mécanique articulé capable d'imiter certaines fonctions humaines telles que la manipulation d'objets ou la locomotion, dans le but de se substituer à l'homme pour la réalisation de certaines tâches matérielles, cette réalisation est plus ou moins autonome selon les facultés de perception de l'environnement dont est doté le robot. [1]

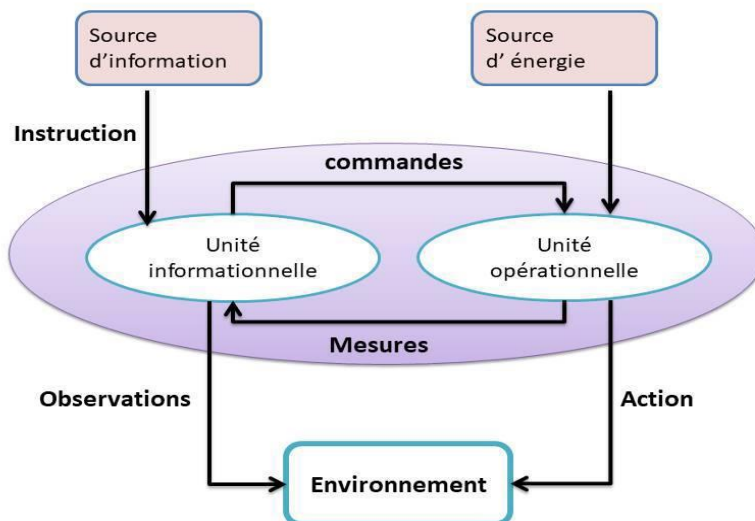


Figure I.3. Structure fonctionnelle d'un robot [4].

I.2.2 Type des robots :

Il existe deux types de robots : robots mobiles et robots manipulateurs.

I.2.2.1 Robots mobiles :

Ce sont des robots capables de se déplacer dans un environnement, Ils sont équipés ou Non de manipulateurs suivant leur utilisation (les robots explorateurs, les robots de services ...) [1].

Chapitre I : État de l'art sur les approches de contrôle en robotique par les techniques de l'IA



Figure.I.4. Robots mobiles [3].

I.2.2.2 Robots manipulateurs :

C'est des robots ancrés physiquement à leur place de travail et généralement mis en place pour réaliser une tâche précise et répétitive.

Ce sont des manipulateurs automatiques programmés qui se substituent à l'homme pour

L'accomplissement de tâches répétitives, (tels que les bras manipulateur, médicaux, les robots industriels...) [1].



Figure.I.5. Robot manipulateur [5].



Figure.I.6 Bras manipulateur parallèle [1].

I.2.3 Domaine d'utilisation des robots :

➤ La robotique industrielle :

L'ISO définit formellement un robot industriel comme le contrôle automatique et reprogrammable de trois axes ou plus. Les applications typiques incluent les robots de peinture et de soudage d'assemblage. Les robots industriels sont largement utilisés dans le domaine automobile.

Chapitre I : État de l'art sur les approches de contrôle en robotique par les techniques de l'IA

➤ **La robotique médicale :**

Un robot médical est un système robotique utilisé dans le cadre d'une application thérapeutique, par exemple lors d'une chirurgie au cours d'un programme de réhabilitation neuromatrice. Du fait des contraintes importantes en matière de sécurité, ce type de robot est en général doté d'un faible niveau d'autonomie.

➤ **La robotique de transport :**

Un système automatique de transport fonctionne de manière automatique, Il peut être guidé dans un parcours contraint (véhicule sur rails) ou plus libre voiture autonome.

➤ **La robotique militaire:**

Un robot militaire, aussi appelé arme autonome, est un robot autonome ou contrôlé à distance, conçu pour des applications militaires. Les drones sont une sous classe des robots militaires.

➤ **La robotique domestique:**

Un robot domestique est un robot de services personnels, utilisés Pour des tâches ménagères. Jusqu'à présent, il y a que rare modèle. Les robots domestiques sont utilisés par exemple en vaisselle, en nettoyage et en cuisine.

I.3 Les techniques classiques de contrôle en robotique :

Les techniques classiques de contrôle en robotique ont été largement utilisées avant l'avènement de l'intelligence artificielle. Voici une description des deux principales approches classiques de contrôle en robotique :

-Contrôle basé sur des modèles physiques : Le contrôle basé sur des modèles physiques repose sur une compréhension approfondie du système robotique et de ses interactions avec l'environnement. Les ingénieurs conçoivent des modèles mathématiques qui décrivent le comportement physique du robot, y compris ses dynamiques, ses contraintes et ses caractéristiques géométriques. En utilisant ces modèles, des lois de contrôle sont développées pour commander les mouvements du robot et atteindre les objectifs souhaités. Cette approche nécessite généralement une connaissance précise des propriétés du robot et de son environnement.

-Contrôle par boucle de rétroaction (feedback) : Le contrôle par boucle de rétroaction est une approche dans laquelle les capteurs du robot fournissent des informations sur l'état actuel du système, qui sont ensuite utilisées pour ajuster les commandes de manière itérative. Le contrôleur mesure l'écart entre l'état désiré et l'état réel du robot, puis génère des commandes

Chapitre I : État de l'art sur les approches de contrôle en robotique par les techniques de l'IA

Pour minimiser cet écart. Ce processus est répété à chaque itération, permettant au robot de s'adapter aux variations de l'environnement et aux perturbations. Les boucles de rétroaction peuvent être conçues de manière simple, comme dans le cas des régulateurs PID (proportionnel-intégral-dérivé), ou plus complexes, en utilisant des méthodes de contrôle avancées telles que les régulateurs prédictifs.

Cependant, ces approches classiques présentent certaines limitations :

- **Sensibilité aux variations du système** : Les techniques classiques de contrôle reposent souvent sur des modèles simplifiés qui peuvent ne pas capturer toutes les nuances et les variations du système robotique réel. Par conséquent, elles peuvent être sensibles aux changements dans les paramètres du robot, de l'environnement ou des tâches.
- **Difficulté à traiter les environnements complexes** : Les approches classiques peuvent avoir du mal à gérer les environnements complexes et imprévisibles, tels que les obstacles mouvants, les surfaces irrégulières ou les interactions avec des objets inconnus. La conception de modèles précis pour de tels scénarios peut être difficile, voire impossible.
- **Besoin d'une programmation manuelle intensive** : Les techniques classiques nécessitent souvent une programmation manuelle approfondie pour spécifier les comportements du robot dans différentes situations. Cela peut être coûteux et laborieux, en particulier pour des tâches complexes nécessitant une adaptation en temps réel.

C'est pourquoi l'intelligence artificielle et l'apprentissage automatique ont gagné en popularité dans le domaine de la robotique, car ils offrent des approches plus flexibles et adaptatives pour le contrôle des robots dans des environnements complexes et variables.

I.4 L'intelligence Artificielle En Robotique :

L'intelligence artificielle est un domaine de l'informatique qui vise à créer des machines capables de simuler des processus cognitifs humains tels que l'apprentissage, la perception, la prise de décision, etc. L'apprentissage automatique (machine Learning) est une sous-branche de l'IA qui permet aux machines d'apprendre à partir de données et d'expériences pour améliorer leurs performances sans être explicitement programmées.

Dans le domaine de la robotique, l'apprentissage automatique est largement utilisé pour développer des algorithmes de contrôle adaptatifs et intelligents. Les réseaux de neurones Artificiels, en particulier les réseaux de neurones profonds (Deep neural networks), sont des architectures d'apprentissage automatique qui ont révolutionné de nombreux domaines,

Chapitre I : État de l'art sur les approches de contrôle en robotique par les techniques de l'IA

Y compris la robotique. Ces réseaux sont capables d'apprendre des représentations complexes et de prendre des décisions basées sur des données d'entrée, ce qui les rend adaptés à des tâches de contrôle en robotique.

I.5 Les techniques d'intelligence artificielle en robotique :

I.5.1 L'apprentissage automatique(ML):

L'apprentissage automatique ou artificiel (Machine Learning en anglais) regroupe tellement de concepts différents et variés qu'il est difficile d'en donner une définition unique. En effet l'apprentissage automatique est pluridisciplinaire et fait intervenir des concepts et des techniques de plusieurs domaines dont il tire ses inspirations et ses outils.

Alors pour ce domaine pluridisciplinaire, nous présentons les définitions suivantes [6] :

❖ **Définition 1:** Du point de vue des machines, on dira qu'une machine apprend dès lors qu'elle change sa structure, son programme ou ses données en fonction de données en entrée ou de réponses à son environnement de sorte à ce que ses performances futures deviennent meilleures.

❖ **Définition 2:** L'apprentissage dénote des changements dans un système qui lui permettent de faire la même tâche plus efficacement la prochaine fois.

❖ **Définition 3:** L'apprentissage automatique fait référence au développement, à l'analyse et à l'implémentation de méthodes qui permettent à une machine (au sens large) d'évoluer grâce à un processus d'apprentissage, et ainsi de remplir des tâches qu'il est difficile ou impossible de remplir par des moyens algorithmiques plus classiques.

❖ **Définition 4:** Un programme informatique apprend à partir de l'expérience E par rapport à une classe de tâches T et une mesure de performance P , si sa performance à l'exécution de tâches de T , mesurée par P , s'améliore avec l'expérience E .

Une synthèse des définitions met en évidence un élément clé : l'apprentissage automatique permet à une machine de s'améliorer par elle-même. Ainsi, l'apprentissage automatique est similaire à l'apprentissage naturel.

I.5.1.1 Méthodes d'apprentissages automatiques :

I.5.1.1.1 L'apprentissage supervisé (Supervised Learning) SL :

Est une tâche d'apprentissage automatique qui consiste à apprendre une fonction qui mappe une entrée à une sortie en se basant sur des paires d'exemples d'entrée-sortie. Il infère une fonction à partir de données d'entraînement étiquetées, qui constituent un ensemble d'exemples d'entraînement. Les algorithmes d'apprentissage supervisé sont des algorithmes qui nécessitent

Chapitre I : État de l'art sur les approches de contrôle en robotique par les techniques de l'IA

une assistance externe. L'ensemble de données d'entrée est divisé en ensembles d'entraînement et de test. L'ensemble d'entraînement contient la variable de sortie qui doit être prédite ou classifiée. Tous les algorithmes apprennent des types de modèles à partir de l'ensemble de données d'entraînement et les appliquent à l'ensemble de données de test pour la prédiction ou la classification. [7]

✚ **Les algorithmes d'apprentissage supervisé** sont utilisés dans diverses applications, telles que la classification d'images, la reconnaissance vocale, la prédiction de prix, la détection de fraude, etc. Ils peuvent être basés sur des méthodes linéaires, des arbres de décision, des réseaux neuronaux, des machines à vecteurs de support (SVM), etc. Les algorithmes d'apprentissage supervisé les plus célèbres ont été discutés ici [7]:

-*Arbre de décision* : Un arbre de décision est un graphe qui représente les choix et leurs résultats sous la forme d'un arbre. Les nœuds du graphe représentent un événement ou un choix, et les arêtes du graphe représentent les règles de décision ou les conditions. Chaque arbre est composé de nœuds et de branches. Chaque nœud représente des attributs dans un groupe qui doit être classé, et chaque branche représente une valeur que le nœud peut prendre

-*Naive Bayes* : est une technique de classification basée sur le théorème de Bayes avec une hypothèse d'indépendance entre les prédicteurs. En termes simples, un classifieur Naive Bayes suppose que la présence d'une caractéristique particulière dans une classe n'a aucun lien avec la présence de toute autre caractéristique. Naive Bayes cible principalement l'industrie de la classification de texte. Il est principalement utilisé pour le regroupement et la classification en fonction de la probabilité conditionnelle de survenue.

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$

↓ ↓ ↓

Likelihood Class Prior Probability

↓ ↓

Posterior Probability Predictor Prior Probability

$$P(c|X) = P(x_1|c) \times P(x_2|c) \times \dots \times P(x_n|c) \times P(c)$$

Figure I.7 : Naive bayes [7].

-*La machine à vecteurs de support (SVM)* : Une autre technique d'apprentissage automatique très utilisée et à la pointe de la technologie est la machine à vecteurs de support (SVM). En apprentissage automatique, les machines à vecteurs de support sont des modèles d'apprentissage

Chapitre I : État de l'art sur les approches de contrôle en robotique par les techniques de l'IA

Supervisé avec des algorithmes d'apprentissage associés qui analysent les données utilisées pour la classification et l'analyse de régression. En plus d'effectuer une classification linéaire, les SVM peuvent effectuer efficacement une classification non linéaire en utilisant ce qu'on appelle la technique du noyau, qui mappe implicitement leurs entrées dans des espaces de caractéristiques de grande dimension. En gros, elles tracent des marges entre les classes. Les marges sont tracées de manière à ce que la distance entre la marge et les classes soit maximale, ce qui permet de minimiser l'erreur de classification.

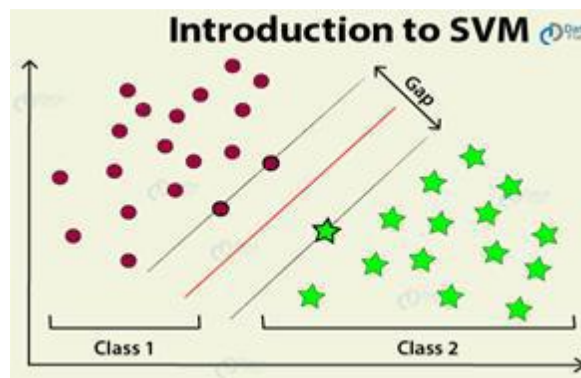


Figure I.8 : Support Vector Machine [7].

I.5.1.1.2 L'apprentissage Non supervisé (Unsupervised Learning):

Les algorithmes non supervisés ne sont pas basés sur des données d'entrée-sortie et considèrent toutes les données comme équivalentes, c'est-à-dire qu'il n'y a pas de variable de sortie spécifique à prédire. Au lieu de cela, ces algorithmes cherchent à découvrir la structure sous-jacente des données en les organisant en groupes ou en clusters. L'objectif est de regrouper des données similaires ensemble, tandis que les données différentes devraient se retrouver dans des groupes distincts. Dans ce contexte, l'apprentissage ne se fait pas en se basant sur des indications fournies par un expert, mais uniquement à partir des motifs et des fluctuations observables dans les données elles-mêmes.

I.5.2 L'apprentissage par renforcement (RL) :

Est un domaine de l'apprentissage automatique (**Reinforcement Learning en anglais**) qui se concentre sur la manière dont les agents logiciels doivent prendre des actions dans un Environnement afin de maximiser une notion de récompense cumulative. L'apprentissage par renforcement est l'un des trois paradigmes fondamentaux de l'apprentissage automatique, aux côtés de l'apprentissage supervisé et de l'apprentissage non supervisé.

Chapitre I : État de l'art sur les approches de contrôle en robotique par les techniques de l'IA

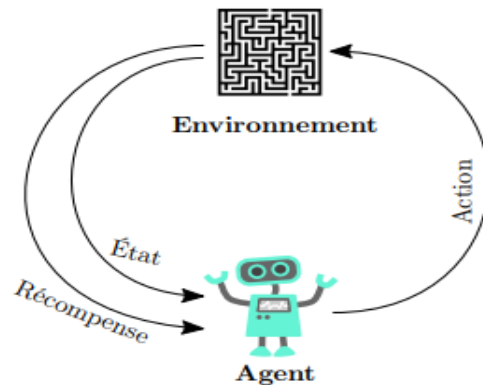


Figure I.9 : Illustration du cadre général de l'apprentissage par renforcement [7].

-L'apprentissage par renforcement utilise en effet diverses techniques pour apprendre des politiques optimales. Voici une brève description de ces techniques :

- 1.Programmation dynamique :** La programmation dynamique est utilisée pour résoudre des problèmes d'optimisation séquentielle. Elle consiste à décomposer un problème en sous-problèmes plus petits et à utiliser des calculs récursifs pour trouver une solution optimale globale. Dans le contexte de l'apprentissage par renforcement, la programmation dynamique peut être utilisée pour résoudre les processus de décision markoviens (MDP), qui décrivent formellement un problème de prise de décision séquentielle.
- 2.Processus de décision markovien (MDP) :** Les MDP fournissent un cadre mathématique pour modéliser des problèmes de prise de décision séquentielle. Ils consistent en un ensemble d'états, d'actions, de transitions probabilistes entre les états, de récompenses associées aux transitions et d'une fonction de valeur qui évalue la qualité d'une politique. Les algorithmes d'apprentissage par renforcement utilisent les MDP pour apprendre et optimiser des politiques.
- 3. Algorithmes de Monte-Carlo :** Les algorithmes de Monte-Carlo sont utilisés pour estimer les valeurs d'état et les valeurs d'action en utilisant des échantillons de séquences d'interaction avec l'environnement. Ils se basent sur l'idée de simuler plusieurs épisodes d'interaction pour estimer Les retours (récompenses cumulatives) associés à chaque état ou action. Les algorithmes de Monte-Carlo permettent de mettre à jour les valeurs d'état et d'action en utilisant des estimations basées sur les retours observés.
- 4.Méthodes de gradient :** Les méthodes de gradient sont utilisées pour apprendre des politiques basées sur des fonctions paramétriques, telles que les réseaux de neurones. Ces méthodes

Chapitre I : État de l'art sur les approches de contrôle en robotique par les techniques de l'IA

optimisent les paramètres de la politique en utilisant des gradients de la fonction de valeur ou de la fonction de log-vraisemblance pour maximiser les récompenses cumulatives attendues.

I.5.2.1 Les algorithmes d'apprentissage par renforcement :

Q-Learning :

Est un algorithme classique d'apprentissage par renforcement basé sur l'estimation des valeurs d'action, également appelées Q-valeurs. Il fonctionne en utilisant une table (ou une fonction d'estimation) appelée la table Q, qui stocke les estimations des récompenses attendues pour chaque paire état-action. L'algorithme itère entre l'exploration (en prenant des actions aléatoires) et l'exploitation (en utilisant les estimations Q pour prendre des actions) pour mettre à jour les Q-valeurs. Les mises à jour sont effectuées en utilisant une règle de mise à jour basée sur la récompense actuelle et les estimations futures des Q-valeurs. Le Q-Learning permet d'apprendre des politiques optimales sans avoir besoin d'un modèle explicite de l'environnement.

SARA (State-Action-Reward-State-Action):

Le SARA est un algorithme d'apprentissage par renforcement qui étend le Q-Learning en incluant des transitions d'état dans le processus d'apprentissage. Contrairement au Q-Learning, qui se concentre sur les paires état-action, le SARA utilise des tuples état-action-récompense-état-action pour apprendre les valeurs Q. Cela permet d'incorporer des informations sur les transitions d'état dans l'apprentissage et peut être particulièrement utile dans les environnements où les états se succèdent de manière séquentielle. L'algorithme SARA utilise également une règle de mise à jour basée sur la récompense actuelle et les estimations futures des Q-valeurs.

DQN (Deep Q-Network) :

Le DQN est un algorithme d'apprentissage par renforcement qui combine le Q-Learning avec des réseaux de neurones profonds. Il a été proposé par Deep Mind et a eu un impact significatif dans le domaine de l'apprentissage par renforcement. Le DQN utilise un réseau de neurones profond pour approximer les valeurs Q. Au lieu d'une table Q, le DQN utilise une fonction d'estimation Q paramétrée par un réseau de neurones. L'apprentissage est effectué en utilisant une combinaison de techniques telles que l'exploration epsilon-greedy, l'expérience de rejeu (replay memory) pour l'échantillonnage des transitions passées, et l'apprentissage par descente de gradient pour minimiser l'erreur entre les estimations Q du réseau et les valeurs cibles basées sur la règle de mise à jour du Q-Learning.

Chapitre I : État de l'art sur les approches de contrôle en robotique par les techniques de l'IA

Le DQN a permis des avancées significatives dans des domaines tels que les jeux Atari et le contrôle de robots. Il a été étendu et amélioré dans de nombreuses variantes, telles que Double DQN, dueling DQN, prioritized experience replay, etc.

I.5.2.2 Avantages Et Inconvénients De l'apprentissage par renforcement :

Avantages	Inconvénients
<ul style="list-style-type: none">• Capacité à apprendre à partir de l'interaction avec l'environnement.• Peut traiter des situations inconnues en utilisant des politiques d'exploration.	<ul style="list-style-type: none">• Nécessite un grand nombre d'itérations d'apprentissage pour obtenir de bons résultats.• Le processus d'apprentissage peut être lent et nécessiter des ressources computationnelles importantes

I.5.3 L'apprentissage profond (Deep Learning) :

Le Deep Learning ou apprentissage profond, permet d'aller plus loin que le machine Learning pour reconnaître des objets complexes comme les images, l'écriture manuscrite, la parole et le langage.

Le Deep Learning exploite des réseaux de neurones multicouches, sachant qu'il en existe de très nombreuses variantes. Le Deep Learning permet aussi de générer des contenus ou d'améliorer des contenus existants, comme pour colorier automatiquement des images en noir et blanc. Il est profond parce qu'il exploite des réseaux de neurones avec de nombreuses couches de filtres. Par contre,

Le Deep Learning n'est pas dédié exclusivement au traitement de l'image et du langage. Il peut servir dans d'autres environnements complexes [8].

➤ Les applications de l'apprentissage automatique dans le contrôle des robots sont diverses et nombreuses. Voici quelques exemples pour illustrer cela :

-Navigation autonome : L'apprentissage automatique peut être utilisé pour permettre aux robots de naviguer dans des environnements complexes et dynamiques, en utilisant des techniques telles que la cartographie, la localisation et la planification de trajectoire.

Chapitre I : État de l'art sur les approches de contrôle en robotique par les techniques de l'IA

- **Manipulation d'objets** : Les robots peuvent apprendre à manipuler des objets de manière sûre et précise en utilisant des algorithmes d'apprentissage automatique pour estimer la position des objets, contrôler les mouvements des bras robotiques, et optimiser les saisies.

- **Reconnaissance d'objets et perception de l'environnement** : Les techniques d'apprentissage automatique peuvent être utilisées pour permettre aux robots de reconnaître et de catégoriser des objets dans leur environnement, de détecter des obstacles, de comprendre des scènes complexes, etc.

- **Apprentissage par imitation** : Les robots peuvent apprendre à exécuter des tâches en imitant des démonstrations humaines, où des mouvements et des comportements humains sont enregistrés et utilisés comme données d'entraînement pour les algorithmes d'apprentissage automatique.

I.5.4 Les réseaux de neurone artificiels:

Depuis quelques années, les réseaux de neurones connaissent une popularité croissante dans divers domaines des sciences de l'ingénieur, y compris celui du génie des procédés. Dans cet article, nous nous concentrons sur l'application des réseaux de neurones pour la commande des processus.

L'un des défis actuels consiste à imiter la nature et à reproduire des modes de raisonnement et de comportement propres à celle-ci. Les réseaux de neurones ont émergé de ce désir et se présentent comme une famille de fonctions non linéaires paramétrées, utilisées dans de nombreux domaines tels que la physique, la chimie, la biologie, la finance, etc. Leur utilisation est particulièrement répandue pour la modélisation de processus et la création de lois de commande.

I.5.4.1 Généralité sur les réseaux de neurones :

Le concept des réseaux de neurones artificiels (ANN - Artificial Neural Network) a été développé dans les années 40, s'inspirant de l'analogie avec le système nerveux humain [9]. Les premières recherches remontent à 1943 et sont le fruit de W. S. McCulloch et W. Pitts [10]. Ils ont postulé que l'impulsion nerveuse est le résultat d'un calcul simple effectué par chaque neurone, et que la pensée émerge grâce à l'effet collectif d'un réseau interconnecté de neurones. Bien que des progrès encourageants aient été réalisés à la fin des années 50, le manque de développement théorique a ralenti ces travaux jusqu'aux années 80 [11].

Chapitre I : État de l'art sur les approches de contrôle en robotique par les techniques de l'IA

I.5.4.1.1 Neurones biologique :

La structure d'un neurone biologique (Fig. I.7) se compose de:

- **Le corps cellulaire**, qui effectue la sommation des influx nerveux qui lui parviennent. Si cette sommation dépasse un certain seuil, il génère un influx nerveux qui se propage à travers l'axone.
- **L'axone**, qui transmet les signaux émis par le corps cellulaire aux autres neurones.
- **Les dendrites**, qui servent de récepteurs principaux du neurone en captant les signaux qui lui sont transmis.
- **Les synapses**, qui permettent aux neurones de communiquer les uns avec les autres en utilisant les axones et les dendrites.

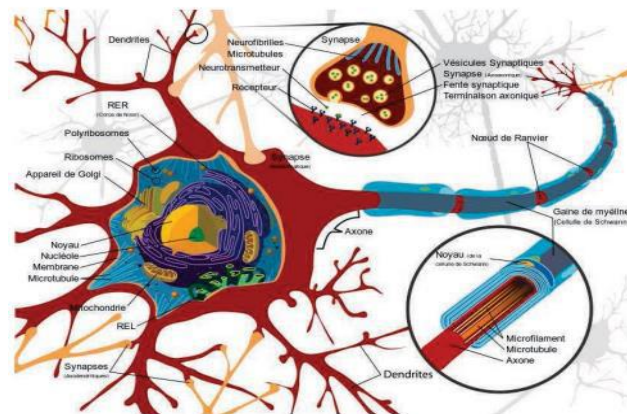


Figure.I.10: Neurone biologique [12].

I.5.4.1.2 Neurone formel (artificiel):

Le neurone formel est un modèle théorique de traitement de l'information qui s'inspire des observations du fonctionnement des neurones biologiques dans le but de reproduire le raisonnement intelligent de manière artificielle [13].

La figure (Fig.I.8) présente la structure d'un neurone artificiel. Chaque neurone artificiel est un processeur élémentaire. Il reçoit un nombre variable d'entrées provenant de neurones "amonts". Chaque entrée est associée à un poids w (weight en anglais) qui représente la force de la connexion. Chaque processeur élémentaire possède une seule sortie qui se divise ensuite pour

Chapitre I : État de l'art sur les approches de contrôle en robotique par les techniques de l'IA

Alimenter un nombre variable de neurones "avals". Chaque connexion est également associée à un poids.

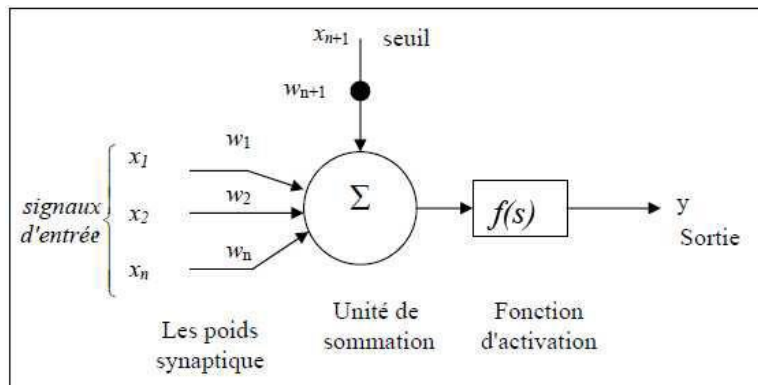


Figure.I.11 : Neurone artificiel [13].

Chaque neurone dans un réseau est connecté à différentes entrées (x_n). Ces entrées peuvent être des variables d'entrée du réseau ou les sorties des couches précédentes. Chacune de ces entrées est associée à un poids (w_i) qui lui est attribué. En ajoutant un biais (β) spécifique à chaque neurone, on calcule un poids total S . Ce poids total est ensuite normalisé dans une plage allant de -1 à 1 pour indiquer le degré d'activation du neurone. Pour propager l'information, une fonction sigmoïde (f) est utilisée, ce qui génère une sortie (y) également normalisée entre -1 et 1 . Toutes ces informations sont stockées dans les valeurs des poids, et un algorithme d'apprentissage par Rétropropagation, basé sur de nombreux exemples, est nécessaire pour optimiser une solution.

Le neurone formel est donc un automate booléen dont la sortie y est telle que [13]:

$$y = \begin{cases} 1 & \text{Si } S > \beta \\ 0 & \text{Si } S < \beta \end{cases} \quad (I.1)$$

Avec :

$$y = f(x) \text{ et } S = \sum_{i=1}^n w_i x_i$$

Dont :

x_i : Représente la i ème entrée du neurone formel.

y : Sortie de neurone.

Chapitre I : État de l'art sur les approches de contrôle en robotique par les techniques de l'IA

β : Seuil du neurone.

W_i : Paramètres de pondération.

f : Fonction de seuillage.

S : Somme pondérée des entrées x_i .

Le tableau ci-dessous montre la mise en correspondance entre un neurone biologique et un neurone Artificiel:

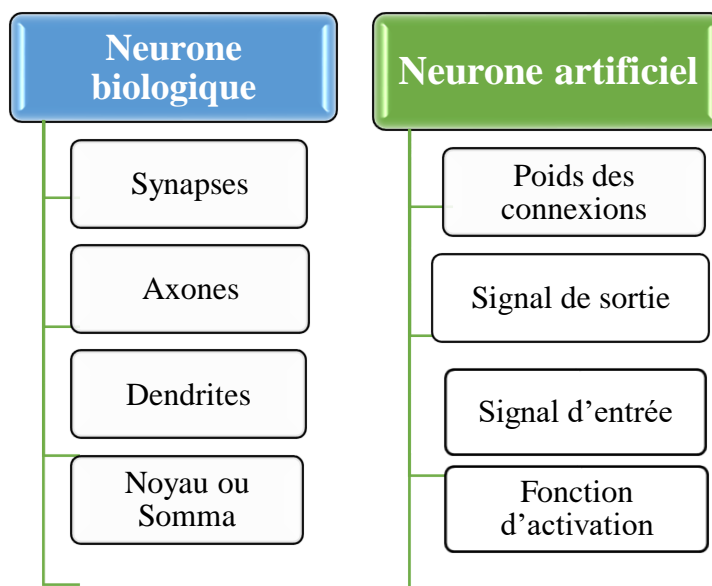


Tableau I.1 : Analogie entre le neurone biologique et le neurone artificiel.

➤ Un neurone formel est défini de manière générale par les éléments suivants [15] :

1. Les entrées du neurone: Elles peuvent être binaires (0, 1) ou réelles, représentant les signaux qui sont transmis au neurone.

2. La fonction d'activation: Cette fonction permet de déterminer l'état interne du neurone en fonction de la somme pondérée de ses entrées. Plusieurs fonctions d'activation couramment utilisées existent, telles que la fonction linéaire et la fonction sigmoïde.

- La fonction linéaire (purelin dans Matlab) : est une fonction d'activation simple où la sortie du neurone est égale à son entrée. Elle ne présente pas de non-linéarité (figure.I.9) Sa fonction est définie par: $f(x) = x$

Chapitre I : État de l'art sur les approches de contrôle en robotique par les techniques de l'IA

-La fonction sigmoïde (logsig dans Matlab) : est une fonction d'activation couramment utilisée. Elle est continue et différentiable, ce qui la rend adaptée à de nombreuses applications. Elle est définie comme une courbe en forme de "S" et est utilisée pour mapper les entrées pondérées du neurone à une sortie comprise entre 0 et 1.

Sa dérivée est simple à calculer. Elle est définie par: $f(x) = \frac{1}{1+e^{-x}}$

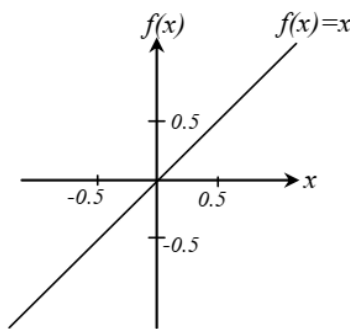


Figure.I.12: Fonction linéaire [16]

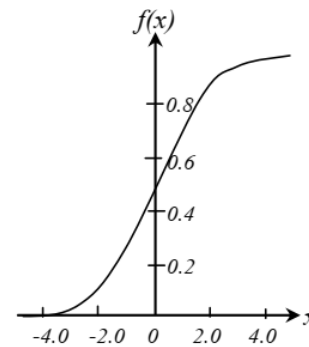


Figure.I.13: Fonction sigmoïde [16].

3. Poids des connexions: Chaque connexion entre les neurones est associée à un poids w_{ij} , où le premier indice i représente le numéro de neurone dans la couche et le deuxième indice j spécifie le numéro de l'entrée. Le poids w_{ij} détermine l'efficacité de la connexion synaptique entre le neurone i et son entrée j . Un poids positif renforce l'entrée, tandis qu'un poids négatif l'inhibe.

En résumé, les poids des connexions dans un neurone artificiel représentent l'importance et l'influence de chaque connexion synaptique. Les poids positifs et négatifs modulent l'impact des entrées sur l'activité et le comportement du neurone.

I.5.4.2 Architecture des réseaux de neurones :

L'architecture d'un réseau de neurones se réfère à la manière dont les neurones sont organisés et connectés au sein du réseau. Cela inclut l'ordre des neurones et les connexions entre eux. Un réseau de neurones typique est composé de plusieurs couches de neurones, allant des entrées aux sorties. On distingue principalement deux types d'architectures de réseaux de neurones : les réseaux de neurones non bouclés (feedforward) et les réseaux de neurones bouclés (récurrent).

I.5.4.2.1 Les réseaux de neurones bouclés :

Appelés aussi réseaux dynamiques ou réseaux récurrents (**RNN pour "Recurrent Neural Networks" en anglais**). Ce sont des réseaux contenant au moins une boucle de contre-réaction sur une de ses couches vers une au moins des couches précédentes (Fig.I.11). Ces réseaux ont une dynamique importante. En effet, à cause de la présence de boucles de contre-réaction et de la réponse non linéaire des neurones, le réseau peut être amené progressivement vers un état stable ou alors, une brusque variation apparaît sur les neurones pour amener le réseau vers un autre état [17].

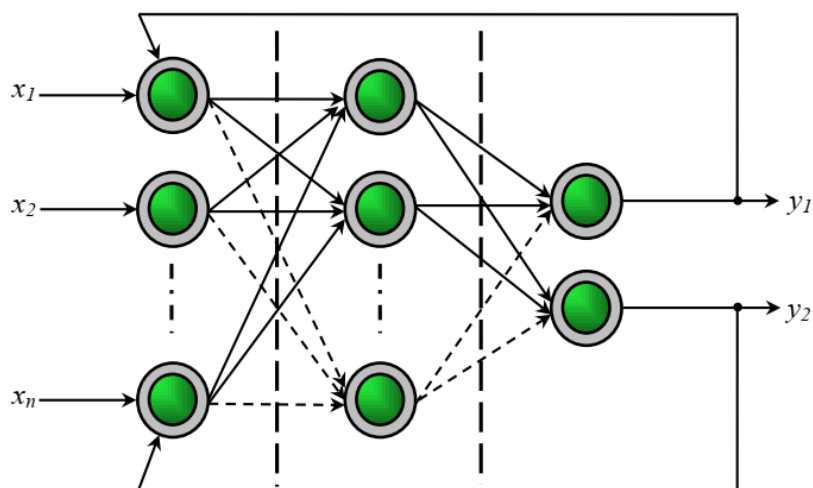


Figure.I.14 : Réseaux de neurone bouclés [17].

I.5.4.2.2 Les réseaux de neurones non bouclés :

Un réseau de neurones non bouclé, également appelé réseau de neurones proactif, réseau statique ou réseau unidirectionnel (**feedforward network en anglais**), est capable d'effectuer une ou plusieurs fonctions algébriques sur ses entrées en composant les fonctions réalisées par chaque neurone individuel [18]. Ce réseau peut être monocouche ou multicouche, avec une connectivité partielle ou complète (tous les neurones d'une couche sont connectés à chaque nœud de la couche suivante) [17].

Graphiquement, ce réseau est représenté par un ensemble de neurones interconnectés où le flux d'information se propage des entrées vers les sorties sans boucle de rétroaction. En se déplaçant dans le réseau à partir d'un neurone quelconque en suivant les connexions, il n'est pas possible de revenir au neurone de départ. Les neurones qui effectuent les derniers calculs de la composition de la fonction sont les neurones de sortie, tandis que les neurones qui effectuent des calculs intermédiaires sont les neurones cachés [18].

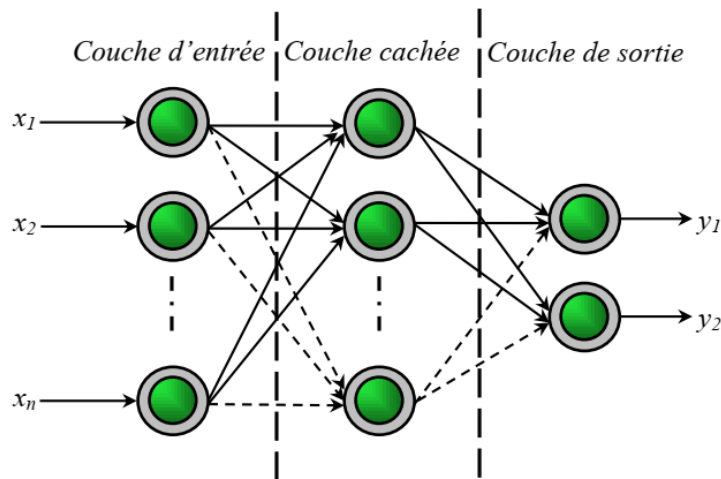


Figure.I.15 : Réseaux de neurone non bouclés [18].

I.5.4.3 L'apprentissage d'un réseau de neurones :

L'information que peut acquérir un réseau de neurones est représentée dans les poids des connexions entre les neurones. L'apprentissage consiste donc à ajuster ces poids de telle façon que le réseau présente certains comportements désirés. En d'autres termes, l'apprentissage des réseaux de neurones consiste à ajuster les poids synaptiques de telle manière que les sorties du réseau (le modèle) soient, pour les exemples (les entrées) présentés au réseau lors de l'apprentissage, aussi proches que possibles des sorties "désirées". Il existe trois types d'apprentissage, un apprentissage supervisé, apprentissage non supervisé et apprentissage par renforcement.

- **L'apprentissage supervisé** : pour lequel on dispose de la sortie désirée et qui consiste à ajuster les poids synaptiques de telle sorte à minimiser l'écart entre la sortie désirée et la sortie du réseau.

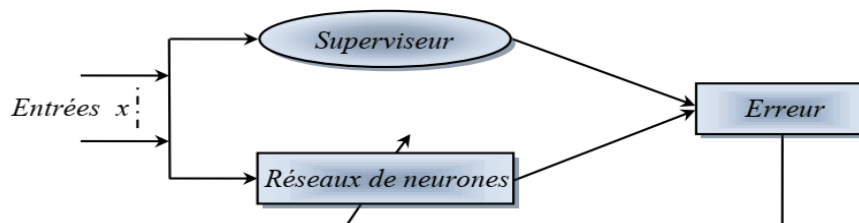


Figure.I.16 : Figuration de l'apprentissage supervisé.

Chapitre I : État de l'art sur les approches de contrôle en robotique par les techniques de l'IA

- **L'apprentissage non supervisé** : pour lequel le réseau de neurones organise lui-même les entrées qui lui sont présentées de façon à optimiser un critère de performances donné.

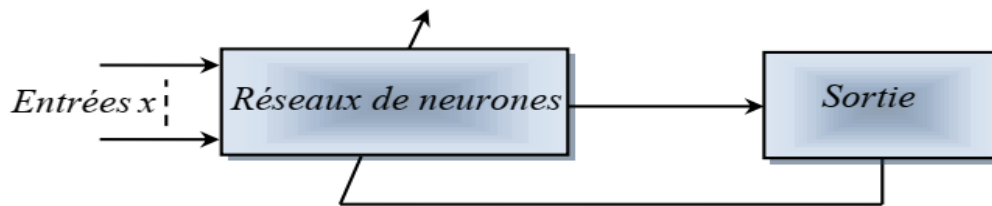


Figure.I.17: Figuration de l'apprentissage non supervisé.

- **L'apprentissage par renforcement** : pour lequel le réseau de neurones est informé d'une manière indirecte sur l'effet de son action choisie. Cette action est renforcée si elle conduit à une amélioration des performances.

I.5.4.4 Règles d'apprentissage des réseaux de neurones :

Les règles d'apprentissage des réseaux de neurones sont des méthodes utilisées pour ajuster les poids des connexions entre les neurones et améliorer les performances du réseau. Voici quelques-unes des règles d'apprentissage les plus couramment utilisées :

-**Rétropropagation du gradient (Backpropagation)** : C'est l'algorithme d'apprentissage le plus répandu pour les réseaux de neurones multicouches. Il utilise la descente de gradient pour ajuster les poids en propageant l'erreur du réseau de sortie vers les couches précédentes, couche par couche. Cela permet de mettre à jour les poids en fonction de l'erreur de prédiction du réseau.

-**Règle de Hebb** : Cette règle d'apprentissage se base sur le principe formulé par Donald Hebb selon lequel "les neurones qui se déclenchent ensemble se lient ensemble". Elle renforce les connexions synaptiques entre les neurones qui sont activés simultanément. Elle est utilisée principalement dans les réseaux de neurones non bouclés pour l'apprentissage de motifs et l'association d'entrées spécifiques avec des sorties souhaitées.

La loi de Hebb s'applique aux connexions entre neurones. Elle s'exprime de la façon suivante: « Si deux cellules sont activées en même temps, alors la force de la connexion augmente ». La modification des poids dépend de l'activation des neurones présynaptiques et post synaptique, ainsi comme le montre le Tableau III.4. x_i et x_j sont respectivement les valeurs d'activation des neurones i et j , ∂w_{ij} (dérivée partielle du poids) correspond à la modification de poids réalisée [19].

Chapitre I : État de l'art sur les approches de contrôle en robotique par les techniques de l'IA

i : Le neurone amont.

j : Le neurone aval.

∂w_{ij} : Le poids de la connexion

x_i	x_j	∂w_{ij}
0	0	0
0	1	0
1	0	0
1	1	+

Tableau.I.2 : La loi de Hebb.

-Règle de Kohonen : Cette règle d'apprentissage est utilisée dans les réseaux de neurones de type "carte auto-organisatrice". Elle permet de regrouper des données similaires en activant les neurones voisins. Les poids des connexions sont ajustés pour rapprocher les neurones qui répondent aux mêmes stimuli.

-Règle de l'apprentissage compétitif : Cette règle d'apprentissage est utilisée dans les réseaux de neurones compétitifs où plusieurs neurones "compétent" pour être activés en réponse à des entrées spécifiques. Les neurones gagnants renforcent leurs connexions synaptiques tandis que les autres connexions sont affaiblies. Cette règle est utilisée pour la catégorisation et la classification des données.

-Apprentissage par essaim de particules (Particle Swarm Optimization) : Cette technique d'apprentissage est inspirée du comportement des essaims d'oiseaux ou de poissons. Les poids du réseau sont traités comme des "particules" qui se déplacent dans un espace de recherche à la recherche d'une solution optimale. Les poids sont mis à jour en fonction de la meilleure performance obtenue par les particules individuelles et du comportement collectif de l'essaim.

I.5.4.5 Différents modèles des réseaux de neurones :

Il existe différents modèles de réseaux de neurones, chacun avec ses propres caractéristiques et architectures spécifiques. Voici quelques-uns des modèles les plus couramment utilisés :

➤ **Perceptron** : Le perceptron est un modèle de réseau de neurones à une seule couche (monocouche) composée de neurones binaires. Il est principalement utilisé pour des problèmes de classification linéaire et utilise la règle du Perceptron pour ajuster les poids des connexions.

Chapitre I : État de l'art sur les approches de contrôle en robotique par les techniques de l'IA

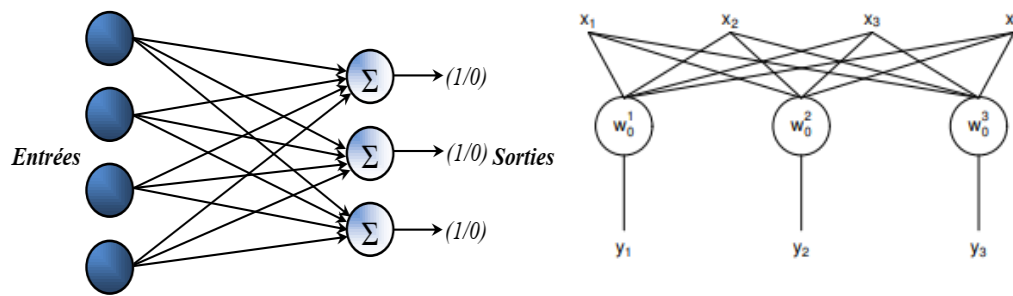


Figure.I.18: Modèle d'un perceptron monocouche à $n = 4$ entrées et $p = 3$ sorties) [20].

➤ **Réseaux de neurones multicouches (Multi-Layer Perceptron, MLP)** : Les réseaux de neurones multicouches sont composés de plusieurs couches de neurones interconnectées, y compris une couche d'entrée, une ou plusieurs couches cachées et une couche de sortie. Ils sont utilisés pour des problèmes de classification et de régression non linéaires et utilisent l'algorithme de Rétro-propagation pour ajuster les poids.

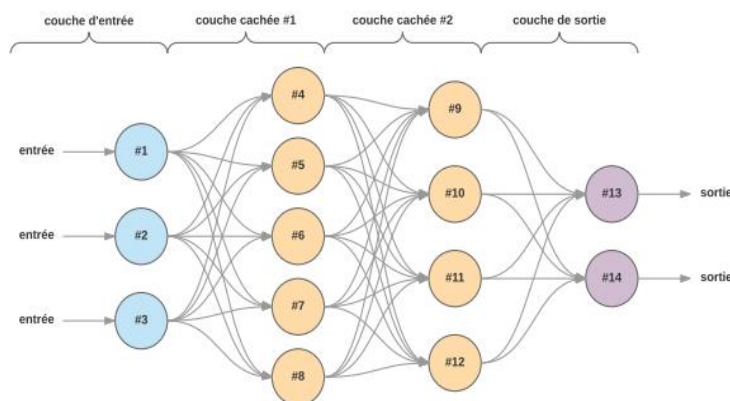


Figure.I.19 : Modèle de perceptron multicouches [21].

➤ **Réseaux de neurones convolutifs (Convolutional Neural Networks, CNN)**
Les réseaux de neurones convolutifs, couramment désignés par l'acronyme CNN sont actuellement considérés comme les modèles les plus performants pour la classification d'images. Ces réseaux comportent deux parties distinctes. Lors de l'entrée d'une image, Celle-ci est représentée sous la forme d'une matrice de pixels. Pour les images en niveaux de gris, la matrice possède deux dimensions. La représentation des couleurs comprend une troisième dimension de profondeur avec une taille de 3 pour représenter les couleurs primaires (Rouge, Vert, Bleu).

Chapitre I : État de l'art sur les approches de contrôle en robotique par les techniques de l'IA

La première partie d'un CNN est la partie convolutive, qui joue un rôle essentiel dans l'extraction des caractéristiques des images. L'image est soumise à une série de filtres, également appelés noyaux de convolution. Ces filtres génèrent de nouvelles images appelées cartes de convolutions. Certains filtres intermédiaires réduisent la résolution de l'image en utilisant une opération de maximum local. À la fin de cette partie, les cartes de convolutions sont aplaties et fusionnées en un vecteur de caractéristiques, connu sous le nom de code CNN.

➤ **Réseaux de neurones récurrents (Recurrent Neural Networks, RNN) :** Les RNN sont conçus pour traiter des données séquentielles et sont dotés d'une mémoire interne qui leur permet de prendre en compte les dépendances temporelles. Ils sont largement utilisés pour des tâches telles que la traduction automatique, la génération de texte et l'analyse de séquences.

➤ **Réseaux de neurones récurrents à mémoire à court terme (Long Short-Term Memory, LSTM) :** Les LSTM sont une variante des RNN qui résolvent le problème du gradient qui disparaît ou explose dans les RNN traditionnels. Les LSTM sont capables de conserver des informations sur de longues séquences et sont largement utilisés dans les tâches de traitement du langage naturel.

➤ **Réseaux de neurones générateurs adversariaux (Generative Adversarial Networks, GAN) :** Les GAN sont des modèles qui utilisent deux réseaux neuronaux en compétition, un générateur et un discriminateur, pour générer des données réalistes. Les GAN sont utilisés pour la génération d'images, la synthèse de voix, la création de musique, etc.

I.5.5 La logique floue :

I.5.5.1 Principe de la logique floue :

La logique floue est une méthode contemporaine qui permet d'exploiter l'expertise d'un opérateur humain qui possède une connaissance approfondie du système à contrôler. Elle offre la capacité de gérer des variables incertaines et imprécises. C'est pourquoi les spécialistes s'efforcent de développer les meilleures stratégies de contrôle qui permettent un réglage optimal des performances. Le principe du réglage par logique floue présente des similitudes avec le raisonnement humain, car les variables manipulées ne sont pas de nature logique (binaire, vrai ou faux), mais plutôt des variables linguistiques qui se rapprochent du langage humain, telles que "un peu chaud", "aller beaucoup plus vite", "freiner à fond", etc. [22].

Chapitre I : État de l'art sur les approches de contrôle en robotique par les techniques de l'IA

I.5.5.2 Notions de base de la logique floue :

➤ **Ensemble ordinaire et ensemble floue :**

Soit un ensemble X continu ou non, fini ou infini, tel que $\langle\langle x \in X \rangle\rangle$ désigne un ensemble de X .

A est un sous ensemble binaire ou ordinaire de X lorsque, $A \subset X$. L'ensemble ordinaire A est défini par sa fonction caractéristique (ou fonction d'appartenance) telle que [23]:

$$\mu_A(x): X \rightarrow [0,1]$$

$$\mu_A(x): \begin{cases} 1 & ; \text{ si } x \in A \\ 0 & ; \text{ sinon} \end{cases}$$

-La notion d'ensemble flou permet des graduations dans l'appartenance d'un élément à une classe, c'est-à-dire autorise un élément à appartenir plus au moins fortement à cette classe. Soit un Référentiel X dénombrable ou non et x un élément de X , alors un ensemble flou A et X est un ensemble de couple tel que :

$$A = \{(x, \mu_A(x))\}$$

$\mu_A(x)$ est le degré d'appartenance de x à A . $\mu_A(x)$ est attribué à X par la fonction d'appartenance

De A qui prend ses valeurs dans un ensemble totalement ordonné. L'ensemble d'appartenance est pris généralement égal à l'intervalle $[0,1]$. La fonction d'appartenance d'un ensemble flou A est définie par:

$$\mu_A : X \rightarrow [0,1]$$

$$x \rightarrow \mu_A(x)$$

➤ **Fonction d'appartenance:** Afin de pouvoir effectuer des calculs numériques sur des variables linguistiques, il est nécessaire de les définir mathématiquement à l'aide de fonctions d'appartenance qui quantifient le degré d'existence de ces variables dans différents sous-ensembles flous [23].

Les représentations graphiques les plus couramment utilisées pour ces fonctions d'appartenance sont illustrées dans la Figure (I.20). Les fonctions d'appartenance sont généralement représentées par des fonctions triangulaires et trapézoïdales, qui sont des fonctions linéaires par morceaux, composées de segments de droite. Ces fonctions sont largement utilisées car elles

Chapitre I : État de l'art sur les approches de contrôle en robotique par les techniques de l'IA

sont simples et permettent de définir des zones où la notion est vraie ou fausse, facilitant ainsi la collecte de l'expertise [24].

En général, ces deux formes sont suffisantes pour délimiter des ensembles flous. Cependant, les formes en cloche et gaussiennes peuvent également être utilisées car elles offrent, dans certains cas, une meilleure représentation des ensembles flous.

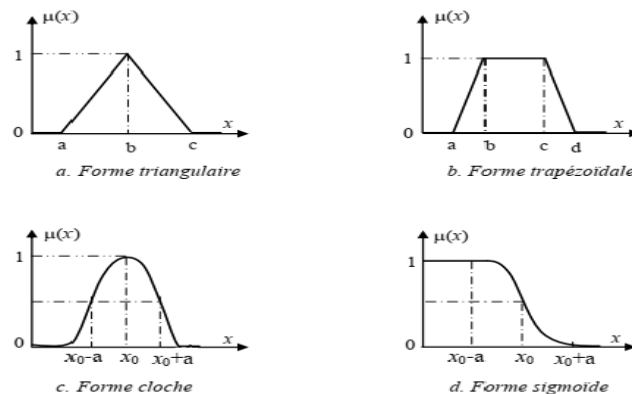


Figure.I.20: Formes usuelles des fonctions d'appartenance [24].

I.5.5.3 Les types de systèmes flous:

Le système d'inférence flou est un système de prise de décision qui repose sur une base de règles. De manière classique, on distingue deux types de systèmes flous basés sur des règles : les systèmes flous à conclusion symbolique (connus sous le nom de systèmes flous linguistiques ou de Mamdani) et les systèmes flous à conclusion fonctionnelle (appelés systèmes flous de Takagi-Sugeno). Dans ces deux types de systèmes flous, les règles sont formulées selon le schéma "SI... ALORS", mais les types de conclusions diffèrent.

◦ Méthode de Mamdani :

Est une approche qui utilise des variables linguistiques pour représenter à la fois les prémisses et les conclusions d'un système flou. Cette méthode se base sur l'utilisation de l'opérateur min (Minimum) pour l'inférence floue et de l'opérateur max (maximum) pour l'agrégation des règles. En pratique, la méthode du centre de gravité est couramment utilisée pour la défuzzification [25]. Pour plus de détails, des informations supplémentaires peuvent être consultées dans les références [26] et [27].

◦ Méthode de Takagi-Sugeno:

La méthode de Takagi-Sugeno est une approche dans laquelle les conclusions sont exprimées sous forme numérique, soit par des constantes (singletons), des polynômes ou des fonctions non

Chapitre I : État de l'art sur les approches de contrôle en robotique par les techniques de l'IA

linéaires des variables d'entrée. L'inférence floue est réalisée en utilisant les opérateurs min ou produit. La valeur finale est obtenue en effectuant une moyenne pondérée des conclusions.

La méthode de Takagi-Sugeno est décrite plus en détail dans la référence [28].

I.5.5.4 L'utilisation de la logique floue :

La logique floue est une technique de résolution de problèmes très puissants avec une Large applicabilité dans le control et la prise de décision. Elle est très utile lorsque le modèle mathématique du problème à traiter n'existe pas ou existe mais difficile à implémenter, ou il est trop complexe pour être évalué assez rapidement pour des opérations en temps réel. Ou bien lorsque des experts humains sont disponibles pour fournir des descriptions subjectives du comportement du système avec des termes en langage naturel.

La logique floue est aussi supposée de travailler dans les situations où il y a de large incertitude et des variations inconnues dans les paramètres et la structure du système [29].

I.5.5.5 Avantages Et Inconvénients De La Logique Floue Et Des Réseaux De Neurones :

<i>Réseaux de neurones</i>	<i>Logique floue</i>
avantages	
<ul style="list-style-type: none">• Le modèle mathématique non requis• Aucune connaissance basée sur les règles• Plusieurs algorithmes d'apprentissage sont Disponibles	<ul style="list-style-type: none">• Le modèle mathématique non requis• La connaissance antérieure sur les règles peut être utilisée• Une interprétation et implémentation simple
inconvénients	
<ul style="list-style-type: none">• Boite noire (manque de traçabilité)• L'adaptation aux environnements différents est difficile et le réapprentissage est souvent Obligatoire (sauf pour le RBF)• la connaissance antérieure ne peut pas être Employée (apprentissage à partir de zéro) (sauf pour le RBF)• Aucune garantie sur la convergence de l'apprentissage	<ul style="list-style-type: none">• Les règles doivent être disponibles• Ne peut pas apprendre• Adaptation difficile au changement de L'environnement• Aucunes méthodes formelles pour l'ajustement

Tableau I.3 : Comparaison entre la logique floue et les réseaux de neurones

I.6 Problèmes et défis de l'utilisation de l'IA dans le contrôle robotique :

◦ Adaptation aux environnements dynamiques :

Les robots évoluent souvent dans des environnements complexes et changeants, tels que des usines ou des espaces publics. L'IA doit être capable de s'adapter rapidement à ces variations et de prendre des décisions en temps réel. Cependant, la modélisation précise et la prédiction des changements environnementaux peuvent être difficiles. De plus, l'IA doit être capable d'interagir de manière sûre avec les humains et de comprendre les intentions et les actions de leur part.

◦ Interprétabilité des modèles d'IA :

Dans de nombreux cas, il est crucial de comprendre comment l'IA prend ses décisions, surtout lorsque des actions sont prises en fonction de ces décisions. Cependant, certains modèles d'IA, tels que les réseaux de neurones profonds, peuvent être difficiles à interpréter. Cela peut rendre difficile la confiance accordée aux décisions prises par le robot. Par conséquent, il est important de développer des méthodes pour rendre les modèles d'IA plus explicables, afin de faciliter la confiance et la compréhension de leur fonctionnement.

◦ Gestion de l'incertitude et des erreurs :

L'IA dans le contrôle robotique doit être capable de gérer l'incertitude inhérente aux données sensorielles et aux modèles prédictifs. Les capteurs peuvent être sujets à des bruits et des imprécisions, et les modèles d'IA ne sont pas parfaits. Par conséquent, il est important de développer des approches qui prennent en compte cette incertitude et qui permettent aux robots de prendre des décisions robustes même en présence d'erreurs. Cela peut impliquer l'utilisation de techniques telles que l'estimation d'état probabiliste, la planification sous incertitude ou l'apprentissage par renforcement avec exploration.

I.7 Conclusion :

En conclusion, ce premier chapitre nous a permis de faire un état de l'art sur les approches de contrôle en robotique utilisant les techniques d'IA. Nous avons exploré plusieurs méthodes, notamment le contrôle par renforcement, l'apprentissage supervisé, les réseaux neurones, l'apprentissage profond, logique floue.

Ces approches offrent des avantages significatifs en permettant aux robots d'acquérir des compétences de contrôle avancé et de s'adapter à des environnements complexes. Le contrôle par renforcement permet aux robots d'apprendre de manière autonome grâce à l'interaction avec leur environnement, tandis que l'apprentissage supervisé leur permet d'imiter des comportements humains ou d'utiliser des données préalablement étiquetées. L'apprentissage profond permet d'extraire des caractéristiques complexes des données et d'apprendre des politiques de contrôle à partir de ces informations.

Ces techniques d'IA ont un impact considérable dans le domaine de la robotique, ouvrant de nouvelles perspectives et possibilités. Cependant, des défis subsistent, tels que la nécessité de grandes quantités de données d'entraînement, la complexité de la conception et du paramétrage des modèles, ainsi que les questions d'interprétabilité et d'éthique.

En somme, les approches de contrôle en robotique par les techniques d'IA ouvrent de nouvelles perspectives pour des robots plus intelligents et capables de réaliser des tâches complexes. Leurs implications sont prometteuses, et nous sommes sur la voie de réaliser des avancées significatives dans ce domaine passionnant.

Chapitre II

État de l'art sur les commandes et les modélisations appliquées aux robots manipulateur

II.1 Introduction

Les robots manipulateurs sont des acteurs clés dans divers domaines tels que l'industrie, la recherche et la robotique médical, en raison de leur capacité à effectuer des tâches complexes et répétitives avec précision. Le chapitre II se penche sur deux éléments fondamentaux qui déterminent leur performance : la commande et la modélisation.

L'objectif de cette étude est de comprendre les mécanismes de commande et de modélisation des robots manipulateurs pour optimiser leur fonctionnement. Nous évaluerons l'impact de différentes approches de commande et de modélisation sur la performance globale des robots, en mettant en évidence les avantages et les limites de chaque méthode.

La commande joue un rôle essentiel en assurant la précision et la répétabilité des mouvements du robot. Une commande efficace permet de minimiser les erreurs de positionnement et d'optimiser ses performances. La modélisation, quant à elle, décrit le comportement du robot de manière mathématique, ce qui facilite la prédiction de son mouvement, la compréhension de sa dynamique et l'anticipation des contraintes.

En combinant une commande bien conçue avec une modélisation précise, les performances du robot manipulateur peuvent être grandement améliorées, garantissant ainsi son bon fonctionnement dans diverses situations.

Ce chapitre examinera en détail les différentes méthodes de commande et de modélisation appliquées aux robots manipulateurs, en mettant en évidence leur importance et leur pertinence dans des contextes pratiques. Enfin, nous aborderons les défis actuels et les perspectives d'avenir pour continuer à améliorer les performances et les capacités des robots manipulateurs.

II.2 Notion générales sur les robots manipulateurs

➤ Définition :

Un RM est un dispositif mécanique poly-articulé et reprogrammable, avec des fonctions similaires à un bras humain. Il est constitué de plusieurs segments articulés permettant de piloter un organe dit terminal ou effecteur, d'où son rôle et de positionner ainsi que l'orienter des matériaux dans le but d'accomplir d'une manière autonome différentes tâches qui sont

Chapitre II : État de l'art sur les commandes et les modélisations appliquées aux robots manipulateur

généralement dangereuses, répétitives, précises et difficiles ou impossible à réaliser par l'être Humain.

Un robot manipulateur doit se conformer aux principes énoncés par les "Trois lois de la robotique" d'Asimov [5]:

Il est interdit à un robot de porter préjudice à un être humain, que ce soit par action directe ou en permettant qu'un tel préjudice survienne du fait de son inaction.

Un robot doit obéir aux ordres donnés par les êtres humains, sauf si ces ordres entrent en conflit avec la première loi.

Un robot doit protéger sa propre existence tant que cette protection n'entre pas en conflit avec la première ou la deuxième loi.

II.2.1 Structure et composants technologiques d'un RM :

Un RM se compose de plusieurs liaisons connectées par des Articulations pour former un bras (figure II.1). Chacune de ces liaisons est articulée par un actionneur Qui est soit électrique, hydraulique ou pneumatique.

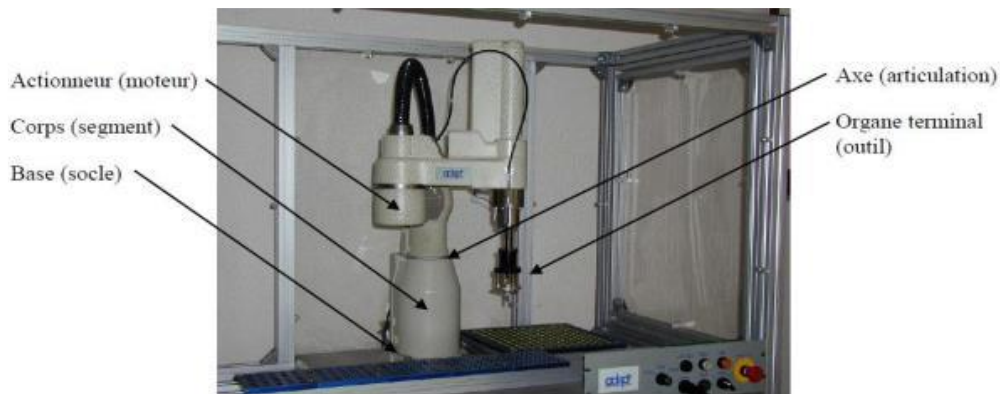


Figure II.1 : La structure d'un robot [30].

La Figure II.2 représente le lien ainsi que les échanges d'informations entre les divers composants d'un robot manipulateur, en mettant en évidence leurs rôles respectifs.

Chapitre II : État de l'art sur les commandes et les modélisations appliquées aux robots manipulateur

- Le robot manipulateur (RM) est composé de deux sous-ensembles : la structure mécanique articulée et l'organe terminal. L'organe terminal offre une interface permettant au RM d'interagir avec son environnement en manipulant divers objets.
- La structure mécanique articulée a pour rôle de positionner et orienter l'organe terminal selon des consignes de position, vitesse et accélération, afin d'atteindre une position géométrique spécifique imposée par la tâche à exécuter. Elle se présente sous forme de chaîne cinématique, constituée de corps rigides reliés par des articulations, qui peuvent être simples, arborescentes, fermées ou parallèles [31].
- Les RM sont équipés d'actionneurs : principalement des moteurs électriques, responsables de mettre en mouvement les différents organes du robot. Dans certains cas, des actionneurs hydrauliques sont utilisés lorsque des efforts importants et des mouvements à vitesse variable sont requis tout en optimisant l'espace et en évitant les étincelles.
- Les capteurs sont également présents sur les RM et sont disposés à divers endroits pour mesurer différentes grandeurs pertinentes. Ils fournissent des informations en temps réel sur l'état du robot, comme sa configuration géométrique (capteurs de position), les efforts qu'il développe (capteurs de courant ou de pression) et même les coordonnées d'un vecteur force ou moment (capteurs de force).
- Le système de commande et de traitement de l'information est le cerveau du robot. Il est composé principalement d'un ordinateur et d'un algorithme de commande. Le système de commande est la partie programmée par l'humain pour que le robot exécute les tâches désirées. Le système de traitement de l'information recueille les données des capteurs et assure un fonctionnement correct du robot en les exploitant efficacement.

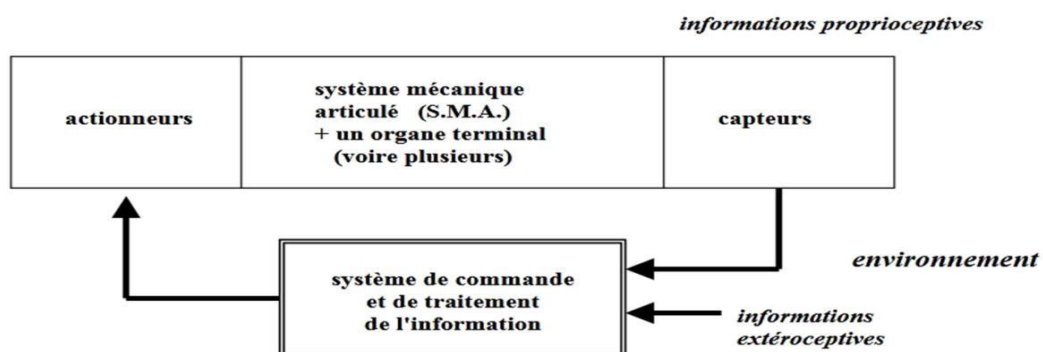
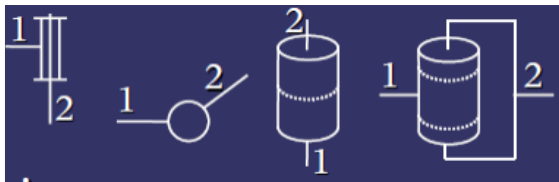


Figure II.2 : La représentation schématique de la composition d'un robot [5].

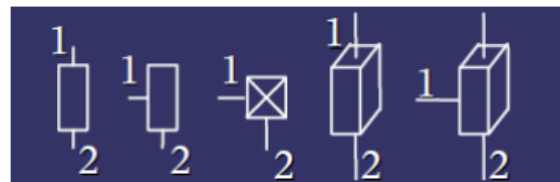
II.2.2 Articulations et degrés de liberté

Articulations:

Une articulation est un mécanisme qui relie deux corps successifs d'un robot, limitant ainsi le nombre de degrés de liberté entre eux, appelé mobilité, qui est généralement représentée par la lettre "m" et se situe dans la plage de 1 à 6. Dans le contexte le plus courant où la mobilité est égale à 1, l'articulation peut être soit rotative (permettant une rotation autour d'un axe) soit prismatique (permettant une translation le long d'un axe) [32].



**Figure II.3.a : Représentation d'une
Articulation rotoïde [30].**



**Figure II.3.b: Représentation d'une
Articulation prismatique [30].**

Degrés de liberté (ddl) :

Le nombre de degrés de liberté d'un robot manipulateur (RM) est équivalent à la somme des degrés de liberté de toutes ses articulations. En d'autres termes, cela représente le nombre de paramètres nécessaires pour décrire pleinement la position et l'orientation d'un élément de la chaîne cinématique par rapport à un autre. Ces degrés de liberté définissent la mobilité du robot, lui permettant de réaliser des mouvements indépendants pour accomplir différentes tâches. Le concept de degrés de liberté est essentiel pour concevoir et contrôler efficacement le robot dans diverses applications sans répéter les descriptions exactes de sources antérieures

II.2.3 Manipulateurs d'architecture sérielle et d'architecture parallèle

Au cours des dernières années, nous avons constaté une forte croissance de l'utilisation des robots dans le secteur industriel, principalement en raison de leur grande flexibilité. Cependant, l'architecture mécanique des robots les plus couramment utilisés est de type sériel, ce qui peut ne pas être idéal pour certaines tâches spécifiques. C'est pourquoi d'autres types d'architectures ont été étudiés et commencent à gagner en popularité dans le domaine de la robotique

Chapitre II : État de l'art sur les commandes et les modélisations appliquées aux robots manipulateur

Industrielle et, plus récemment, dans celui de la machine-outil. Parmi ces architectures alternatives, on retrouve les manipulateurs d'architecture parallèle.

II.2.3.1 Les manipulateurs d'architecture sérielle :

Un robot manipulateur d'architecture sérielle est composé d'une chaîne cinématique simple dans laquelle la base et l'organe effecteur sont reliés à un seul corps, ce qui signifie qu'ils ont un degré de connexion égal à un. Les autres éléments de la chaîne ont un degré de connexion égal à deux, car ils sont reliés à deux corps. La Figure II.4 représente la structure d'un robot manipulateur d'architecture sérielle composé de $n + 1$ corps, notés C_0, \dots, C_n , et de n articulations. Le corps C_0 représente la base du mécanisme, tandis que l'organe terminal est attaché au corps C_n .

Chaque articulation j connecte le corps C_j au corps C_{j+1} , et chaque corps C_j a son propre repère. On peut utiliser le paramétrage de Denavit-Hartenberg pour définir les paramètres cinématiques de ce type d'architecture. [33]

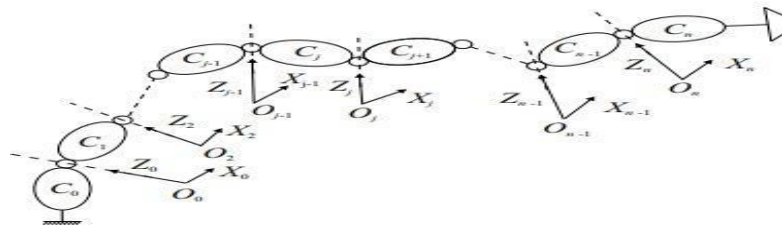


Figure II.4 : Structure d'un manipulateur d'architecture sérielle [33].

II.2.3.2 Les manipulateurs d'architecture parallèle :

Les manipulateurs d'architecture parallèle sont composés d'une plate-forme mobile avec plusieurs degrés de liberté et d'une base fixe, reliées par des chaînes cinématiques fermées indépendantes. Une chaîne cinématique fermée est une série de liaisons mécaniques dans laquelle l'un des membres, autre que la base, possède trois degrés de connexion ou plus. Dans la figure II.5, les corps C_1, C_2, C_3, C_4 et C_5 forment une chaîne cinématique fermée. [33]

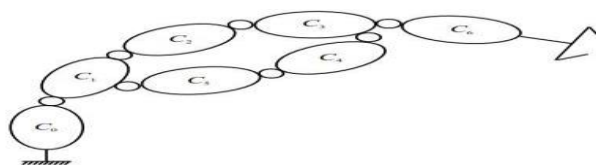


Figure II.5 : Représentation d'une chaîne cinématique fermée [4].

Chapitre II : État de l'art sur les commandes et les modélisations appliquées aux robots manipulateur

- Les manipulateurs d'architecture parallèle ont l'avantage d'être rigides, précis et capables de transporter des charges lourdes. Cependant, leur espace de travail est généralement limité. En variant le nombre et la topologie des chaînes cinématiques du manipulateur, on peut créer un grand nombre de configurations différentes pour les manipulateurs d'architecture parallèle.

La figure II.6 représente, quant à elle, une machine d'usinage d'architecture parallèle de type Delta, dotée de trois degrés de liberté de translation. Ces architectures peuvent être utilisées, par exemple, pour des opérations de fraisage, de soudage au laser, d'assemblage ou de transfert.

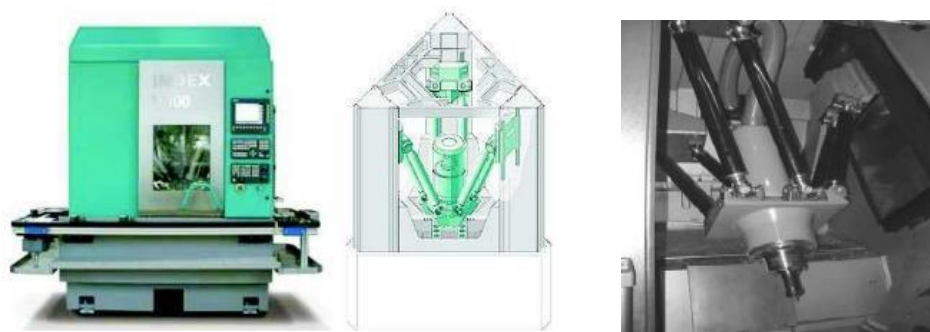


Figure II.6 : Machine parallèle de type Delta [34].

Pour choisir le type de manipulateur adapté à une application donnée, il est essentiel de comparer les propriétés des manipulateurs d'architecture sérielle et d'architecture parallèle. Cette comparaison est résumée ci-dessous [33].

Manipulateurs d'architecture sérielle	Manipulateurs d'architecture parallèle
<ul style="list-style-type: none"> =Succession de segments en série de la base vers l'effecteur =Chaîne cinématique ouverte +Important espace de travail - Faible précision - Faible charge transportable - Faible raideur 	<ul style="list-style-type: none"> =Tout segment au contact de la base et de l'effecteur =Chaines cinématiques fermées - Espace de travail restreint (Précision non définie) + Lourde charge transportable (raideur non définie)

Tableau II.1 – Comparaison des manipulateurs d'architectures sérielle et parallèle

II.2.4 Classification géométrique des robots manipulateurs

On peut classer les robots suivant leur configuration géométrique, autrement dit L'architecture de leur porteur. [35]

○ **Structure cartésienne (PPP) :**

Parmi les configurations de robots, celle à trois liaisons prismatiques est la plus ancienne sur le plan historique. Elle trouve son origine de manière logique dans la conception traditionnelle des machines-outils à trois axes, telles que les rectifieuses ou les fraiseuses. Cependant, cette structure est relativement peu utilisée, sauf dans certaines applications spécifiques, notamment les robots pratiques, les robots de magasinage, et d'autres applications particulières.

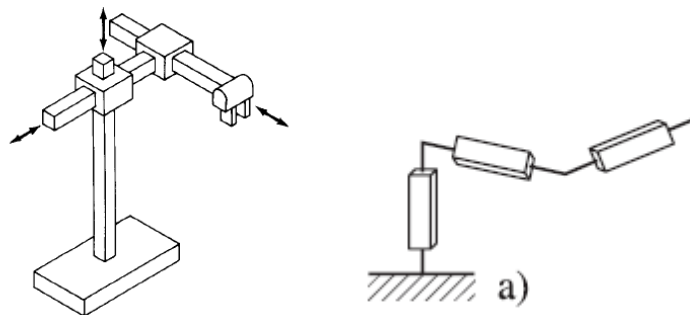


Figure II.7: Robot cartésien [36].

○ **Structure sphérique :**

Elle est une structure de robots à trois axes avec un espace de travail sphérique. Elle est caractérisée par sa grande charge utile, par contre sa rigidité mécanique est inférieure à celle des deux structures précédentes.

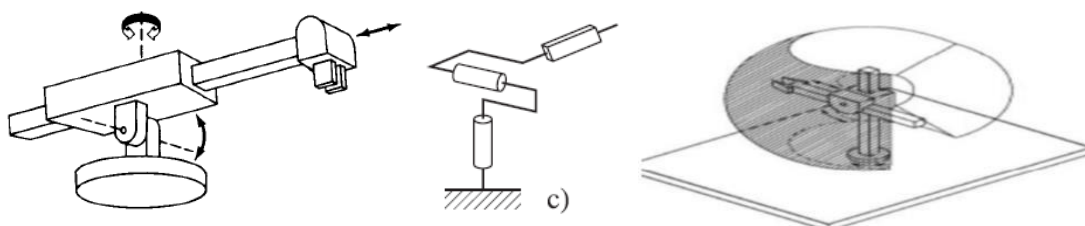


Figure.II.8 : Structure d'un bras manipulateur sphérique [36].

Chapitre II : État de l'art sur les commandes et les modélisations appliquées aux robots manipulateur

○ La structure cylindrique (RPP) ou (PRP) :

Associe une rotation et deux translations. Elle présente l'inconvénient d'offrir un volume de travail faible devant un encombrement total important. Elle n'est pratiquement plus utilisée.

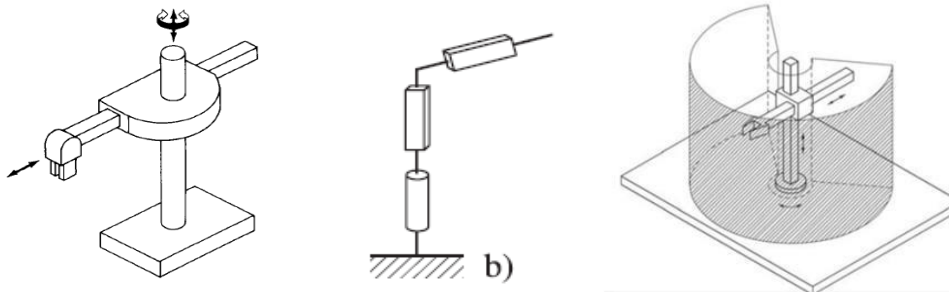


Figure II.9: Robot cylindrique [36].

○ La structure SCARA :

A axes de rotation parallèles est l'une des plus utilisées, en particulier pour des tâches de maintenance ou d'assemblages très fréquents dans l'industrie. Ce succès commercial est lié au fait que le ratio entre le volume de travail et l'encombrement est très favorable et aussi que la structure SCARA est très adaptée à ce type de tâches.

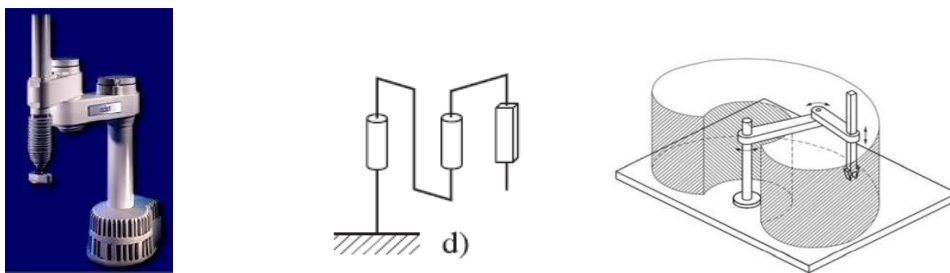


Figure II.10 Robot SCARA [36].

○ La structure 3R (anthropomorphe) :

Permet d'amener un solide en un point de l'espace par trois rotations, généralement une à axe vertical et deux à axes horizontaux et parallèles c'est le porteur « généraliste par excellence, pouvant se programmer facilement pour différents types de tâches et disposant d'un volume de travail conséquent.

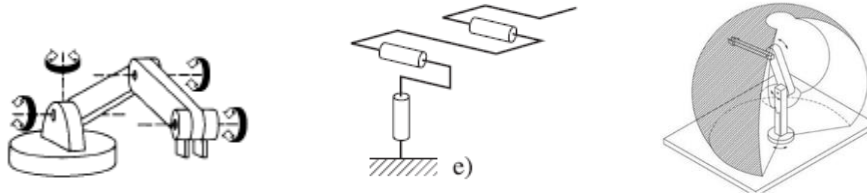


Figure II.11: Robot 3R [36].

II.3 Modélisation des robots manipulateurs

II.3.1 Modélisation géométrique :

La conception et le contrôle d'un robot nécessite la formulation de divers modèles mathématiques essentiels. Parmi eux, les modèles de transformation entre l'espace des articulations et l'espace opérationnel revêtent une importance primordiale. En effet, les robots sont pilotés dans l'espace des articulations, tandis que les tâches à accomplir sont définies dans l'espace opérationnel.

Ces modèles de transformation sont classés en deux catégories fondamentales:

-Modèles de géométrie directe et inverse : Ces modèles établissent la corrélation entre la position et l'orientation de l'organe effecteur et les paramètres articulaires du système mécanique. Ils permettent de déterminer la position de l'organe effecteur en fonction des valeurs des paramètres articulaires, ainsi que le calcul inverse associé.

-Modèles de cinématique directe et inverse : Ces modèles décrivent la relation entre la vitesse de déplacement de l'organe effecteur et les vitesses des articulations. Ils permettent de prédire la vitesse de l'organe effecteur en fonction des vitesses des articulations, et vice versa.

Les modèles dynamiques établissent les liens entre les efforts exercés sur les entrées ou les forces agissant sur les actionneurs et les paramètres tels que les positions, les vitesses et les accélérations des articulations. [37] [38]

II.3.1.1 Transformation homogène et coordonnées généralisées :

1. Coordonnées homogènes :

Soit $(P_x; P_y; P_z)$ les coordonnées cartésiennes d'un point P arbitraire, mesuré dans le repère R_i $(O_i; X_i; Y_i; Z_i)$, Les coordonnées homogène du point P sont $(\omega.P_x; \omega.P_y; \omega.P_z; \omega)$ ou ω est Le facteur d'échelle dans la robotique $\omega = 1$. Les coordonnées homogènes du point P sont représentées par le vecteur colonne [39] :

Chapitre II : État de l'art sur les commandes et les modélisations appliquées aux robots manipulateur

$$\mathbf{p} = \begin{pmatrix} p_x \\ p_y \\ p_z \\ 1 \end{pmatrix} \quad (\text{II.1})$$

2. Transformation homogène :

La compréhension de la transformation de repère se révèle essentielle, car elle offre plusieurs avantages clés :

- Elle permet de représenter la configuration relative des composants du robot entre eux.
- Elle offre la capacité de déterminer les positions requises pour l'organe terminal en utilisant son propre repère.
- Elle facilite la description et la gestion des forces impliquées lorsque le robot interagit avec son milieu environnant.

❖ Transformations homogènes des repères :

La transformation (translation et/ou rotation) d'un repère R_i au repère R_j est représentée

Par la matrice de transformation homogène de dimension (4*4) [39] :

$${}^i_jT = [{}^i_jS \quad {}^i_jn \quad {}^i_ja \quad {}^i_jp] = \begin{bmatrix} s_x & n_x & a_x & p_x \\ s_y & n_y & a_y & p_y \\ s_z & n_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{II.2})$$

i_jS , i_jn et i_ja Sont les vecteurs unitaires des axes $X_j; Y_j; Z_j$ du repère R_j , exprimé dans R_i

i_jp vecteur exprimant l'origine du repère R_j dans le repère R_i .

On peut dire également que la matrice i_jT définit le repère R_j dans le repère R_i , comme on peut noter la matrice de transformation sous forme partitionnée :

$${}^i_jT = \begin{bmatrix} 0 & {}^i_jA & 0 & {}^i_jP \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} {}^i_jS & {}^i_jn & {}^i_ja & {}^i_jp \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{II.3})$$

Finalement, la matrice de transformation i_jT :

- Est considéré comme la représentation du repère R_j dans le repère R_i .

Chapitre II : État de l'art sur les commandes et les modélisations appliquées aux robots manipulateur

➤ Permet le passage du repère R_i au repère R_j

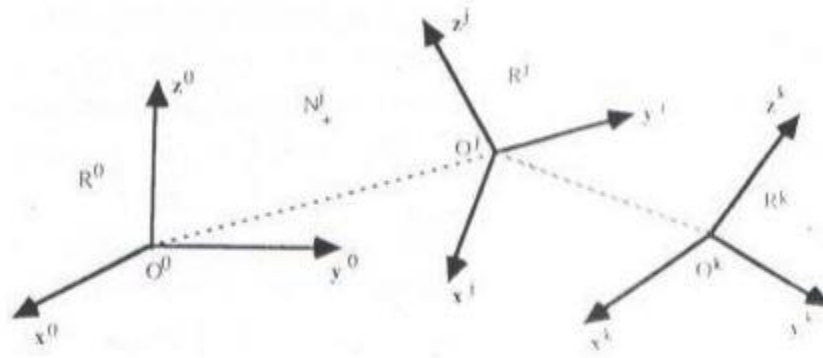


Figure II.12 : Transformation des repères [4].

❖ Matrice de translation pure homogène :

Soit **Trans** (a, b, c) la transformation d'une translation pure où a, b et c sont les translations le long des axes **x**, **y** et **z** respectivement.

La matrice de transformation de cette translation a la forme suivante :

$${}^i_j T = \text{Trans}(a, b, c) = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{II.4})$$

Propriétés :

- $\text{Trans}(a, b, c) = \text{Trans}(x, a) \text{Trans}(y, b) \text{Trans}(z, c)$.
- L'ordre des multiplications étant quelconque.

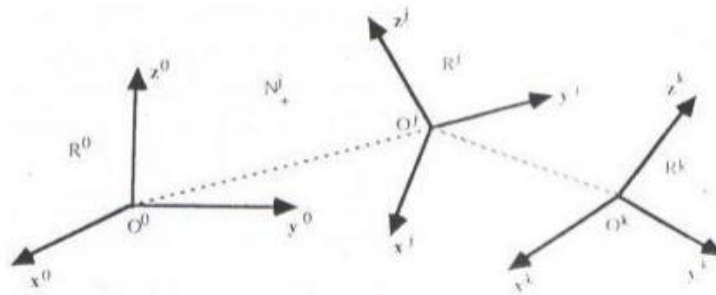


Figure II.13 : Transformation de translation pure [4].

❖ Matrice de rotation homogène :

On définit $\text{Rot}(x, \theta)$, $\text{Rot}(y, \theta)$, $\text{Rot}(z, \theta)$ la transformation homogène qui s'exprime par [37] :

$${}^i_j T = Rot(x; \theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta & 0 \\ 0 & \sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (II.5)$$

$${}^i_j T = Rot(y; \theta) = \begin{bmatrix} \cos \theta & 0 & \sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (II.6)$$

$${}^i_j T = Rot(z; \theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (II.7)$$

II.3.1.2 Modèle géométrique direct:

Représente l'ensemble des relations géométriques permettant le calcul des coordonnées opérationnelles en fonction des coordonnées articulaires. En d'autres termes, ce modèle permet de décrire la situation (position et/ou orientation) de l'organe terminal à partir des mesures de positions articulaires. Dans le cas d'une chaîne ouverte simple, il peut être représenté par la matrice de transformation T_n^0 [4] :

$$T_n^0 = T_1^0(q_1) T_2^1(q_2) T_3^2(q_3) \cdots T_n^{n-1}(q_n) \quad (II.8)$$

Le MGD du robot peut aussi être représenté par la relation :

$$X = f(q) \quad (II.9)$$

• q : est le vecteur des variables articulaires tels que :

$$q = [q_1 q_2 q_3 \cdots q_n]^T \quad (II.10)$$

Les coordonnées opérationnelles sont définies par :

$$X = [X_1 X_2 \cdots X_m]$$

II.3.1.2.1 Paramètres de Denavit-Hartenberg :

Une structure ouverte simple est composée de $n+1$ corps notés C_0, \dots, C_n et de n articulations. Le corps C_0 désigne la base du robot et le corps C_n le corps qui porte l'organe terminal. L'articulation i relie le corps C_i au corps C_{i-1} [40]:

Chapitre II : État de l'art sur les commandes et les modélisations appliquées aux robots manipulateur

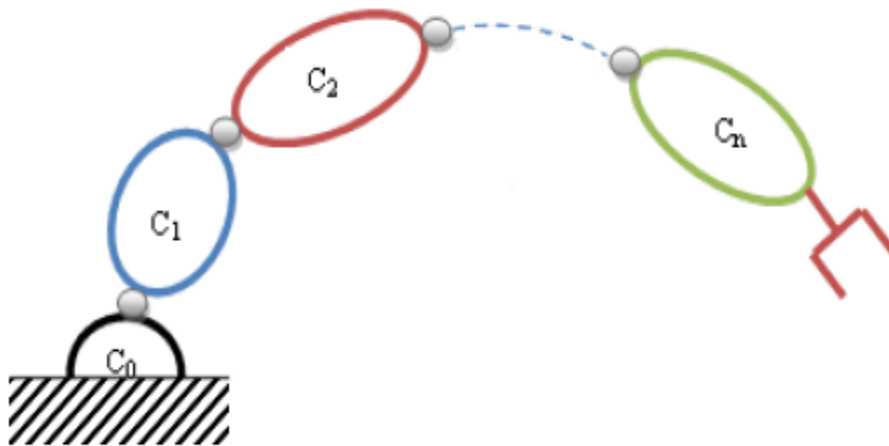


Figure II.14 : Robot à structure ouverte simple [37].

✚ La méthode de description est fondée sur les règles et conventions suivantes :

- Les corps sont supposés parfaitement rigides. Ils sont connectés par des articulations considérées comme idéales (pas de jeu mécanique, pas d'élasticité), soit rotoïdes, soit prismatiques ;

- le repère $R_i (x_i ; y_i ; z_i)$ est lié au corps C_i tels que :

- L'axe Z_i est porté par l'axe de l'articulation i .

- L'axe X_i est porté par la perpendiculaire commune aux axes Z_i et Z_{i-1}

- L'axe Y_i est formé par la règle droite pour compléter les coordonnées du système $(x_i ; y_i ; z_i)$.

➤ **Les paramètres de Denavit-Hartenberg :**

La matrice de transformation du repère R_{i-1} au repère R_i s'exprime en fonction des paramètres (de Denavit-Hartenberg) suivants (Fig. II.15) :

α_i : L'angle de rotation entre l'axe Z_{i-1} et l'axe Z_i autour de X_{i-1} .

d_i : La distance entre l'axe Z_{i-1} et l'axe Z_i le long de l'axe X_{i-1} .

θ_i : L'angle de rotation entre les axes X_{i-1} et X_i autour de Z_i .

r_i : la distance entre les axes X_{i-1} et X_i le long de l'axe Z_i .

Chapitre II : État de l'art sur les commandes et les modélisations appliquées aux robots manipulateur

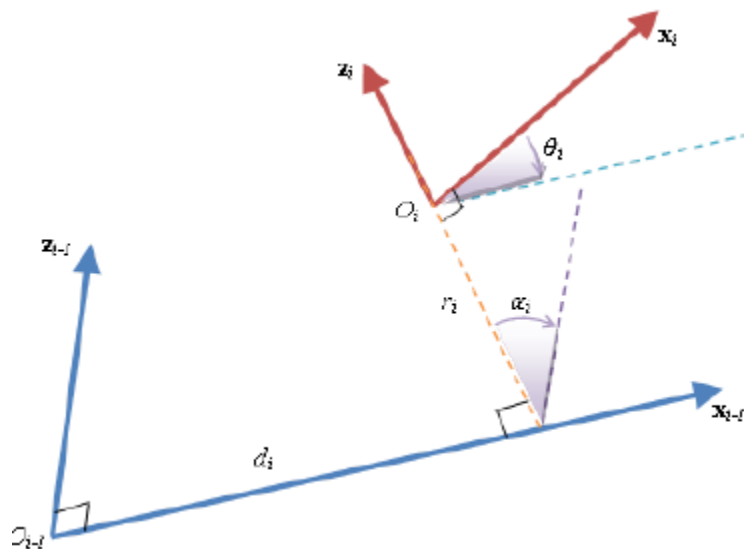


Figure II.15: Paramètres de Denavit-Hartenberg [37].

○ La variable articulaire q_i associée à la $i^{\text{ème}}$ articulation est soit θ_i , soit r_i , selon que cette articulation est de type rotoïde ou prismatique, ce qui se traduit par la relation [30] :

$$q_i = \bar{\sigma}_i \theta_i + \sigma_i r_i \quad (\text{II.11})$$

Avec :

$\sigma_i = 0$ Si l'articulation i est rotoïde

$\sigma_i = 1$ Si l'articulation i est prismatique.

$$\bar{\sigma}_i = 1 - \sigma_i$$

De la même façon on définit la variable \bar{q}_i :

$$\bar{q}_i = \sigma_i \theta_i + \bar{\sigma}_i r_i \quad (\text{II.12})$$

➤ La matrice de transformation définissant R_i dans R_{i-1} est donnée par [40]:

$$T_i^{i-1} = Rot(x, \alpha_i) \times Trans(x, d_i) \times Rot(z, \theta_i) \times Trans(z, r_i) \quad (\text{II.13})$$

$$T_i^{i-1} = \begin{bmatrix} \cos \theta_i & -\sin \theta_i & 0 & d_i \\ \cos \alpha_i \sin \theta_i & \cos \alpha_i \cos \theta_i & -\sin \alpha_i & -r_i \sin \alpha_i \\ \sin \alpha_i \sin \theta_i & \sin \alpha_i \cos \theta_i & \cos \alpha_i & r_i \cos \alpha_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{II.14})$$

Chapitre II : État de l'art sur les commandes et les modélisations appliquées aux robots manipulateur

II.3.1.3 Modèle géométrique inverse :

Il permet à calculer les coordonnées articulaires correspondant à une situation donnée de l'organe terminal (c'est-à-dire les coordonnées opérationnelles ou cartésiennes).

$$\mathbf{q} = \mathbf{f}^{-1}(\mathbf{X})$$

On peut distinguer trois méthodes de calcul du MGI [37] [41].

- La méthode de Paul qui traite séparément chaque cas particulier et convient pour la plupart des robots industriels,
- La méthode de Pieper, qui permet de résoudre le problème pour les robots à six degrés de liberté avec trois articulations rotoïde ou trois articulation prismatiques.
- La méthode générale de Raghavan et Roth, donnant la solution générale des robots à six articulations à partir d'un polynôme de degré au plus égal à 16.

II.3.1.3.1 Calcul du modèle géométrique inverse par la méthode de Paul :

Soit la matrice de transformation homogène d'un robot manipulateur (II.8) :

$$T_n^0 = T_1^0(q_1) T_2^1(q_2) T_3^2(q_3) \cdots T_n^{n-1}(q_n) \quad (\text{II.15})$$

Soit U_0 la situation désirée telle que :

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

On cherche à résoudre le système d'équation s suivant :

$$U_0 = T_1^0(q_1) T_2^1(q_2) T_3^2(q_3) \cdots T_n^{n-1}(q_n) \quad (\text{II.16})$$

Pour résoudre le système (2.16), Paul a proposé une méthode (**Méthode de Paul**) qui consiste à pré-multiplier successivement les deux membres de l'équation par les matrices T_{i-1}^i pour i de 1 à $n - 1$. Ces opérations permettent d'isoler les variables d'articulations l'une après l'autres.

$$\begin{aligned} U_0 &= T_1^0 T_2^1 T_3^2 \cdots T_n^{n-1} \\ T_0^1 U_0 &= T_2^1 T_3^2 T_4^3 \cdots T_n^{n-1} \\ T_1^2 U_1 &= T_3^2 T_4^3 \cdots T_n^{n-1} \\ T_2^3 U_2 &= T_4^3 T_5^4 \cdots T_n^{n-1} \\ &\vdots \\ T_{n-2}^{n-1} U_{n-2} &= T_n^{n-1} \end{aligned}$$

Chapitre II : État de l'art sur les commandes et les modélisations appliquées aux robots manipulateur

Avec :

$$U_{j+1} = T_n^{j+1} = T_j^{j+1} U_j, \text{ pour } j = 0, \dots, n$$

Exemple :

Pour un robot à six degrés de liberté on procède comme suit :

$$U_0 = T_1^0 T_2^1 T_3^2 T_4^3 T_5^4 T_6^5$$

$$T_0^1 U_0 = T_2^1 T_3^2 T_4^3 T_5^4 T_6^5$$

$$T_1^2 U_1 = T_3^2 T_4^3 T_5^4 T_6^5$$

$$T_2^3 U_2 = T_4^3 T_5^4 T_6^5$$

$$T_3^4 U_3 = T_5^4 T_6^5$$

$$T_4^5 U_4 = T_6^5$$

II.3.2 Modélisations Cinématique :

De manière générale, la cinématique se focalise sur le mouvement en ne prenant pas en compte les forces qui le génèrent ; elle se penche sur des aspects tels que la position, la vitesse et l'accélération.

En ce qui concerne la modélisation cinématique d'un robot manipulateur, elle établit une correspondance entre la vitesse de déplacement de la sonde sur le corps du patient et la vitesse des articulations \dot{q}_i pour $(i=1\dots n)$ [35]:

II.3.2.1 Modèle Cinématique direct :

Le modèle cinématique directe (MCD) du robot décrit les variations élémentaires des coordonnées opérationnelles en fonction des variations élémentaires des coordonnées articulaires. Ce MCD est donnée par la relation suivante [42, 4] :

$$\dot{X} = J(q)\dot{q} \quad (\text{II.17})$$

Où $j(q)$ désigne la matrice jacobéenne de dimension $(m \times n)$ du mécanisme égale à $\frac{\partial X}{\partial q}$ et fonction de la configuration articulaire q . La matrice jacobienne intervient dans le calcul du modèle différentielle direct qui donne les variations élémentaires dX des coordonnées opérationnelles en fonction des variations élémentaires des coordonnées articulaires dq , soit :

Chapitre II : État de l'art sur les commandes et les modélisations appliquées aux robots manipulateur

$$(dX) = J(q)(dq) \quad (II.18)$$

- Elle forme le fondement du modèle différentiel inverse, ce qui autorise le calcul d'une solution locale pour les variables articulaires q en fonction des coordonnées opérationnelles X .
- Dans des situations statiques, la jacobienne est employée pour établir la corrélation entre les forces et les couples appliqués par l'organe terminal sur l'environnement, et les forces ainsi que les couples produits par les actionneurs.
- Elle simplifie la détermination des singularités et de la dimension de l'espace opérationnel atteignable du robot.

▪ Calcul de la matrice jacobienne par dérivation du MGD :

Le calcul de la matrice Jacobienne peut se faire en dérivant le MGD, $X = F(q)$, à partir de la relation suivante :

$$J_{ij} = \frac{\partial F_i(q)}{\partial q_j} \quad i=1, \dots, m; j=1, \dots, n \quad (II.19)$$

Où J_{ij} l'élément (i, j) de la matrice Jacobienne J.

Cette méthode est facile à mettre en œuvre pour des robots à deux ou trois degrés de liberté.

▪ Matrice jacobienne cinématique :

La matrice jacobienne peut être dérivée par une méthode de calcul directe qui repose sur la corrélation entre les vecteurs de vitesse de translation et de rotation V_n et W_n du repère R_n , ainsi que les vitesses des articulations \dot{q} :

$$\begin{bmatrix} V_n \\ W_n \end{bmatrix} = J_n \dot{q} \quad (II.20)$$

Cette équation peut être exprimée comme :

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ c \\ w_y \\ w_z \end{bmatrix} = \begin{bmatrix} j_{l,1} & j_{l,2} & \dots & j_{l,n} \\ j_{A,1} & j_{A,2} & \dots & j_{A,n} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \vdots \\ \dot{q}_n \end{bmatrix} \quad (II.21)$$

Chapitre II : État de l'art sur les commandes et les modélisations appliquées aux robots manipulateur

Ou \dot{x} , \dot{y} et \dot{z} sont les dérivés par rapport au temps des coordonnées x , y et z de l'origine du référentiel de l'outil par rapport au référentiel de l'atelier, et w_x , w_y et w_z ne sont pas les dérivées d'une représentation quelconque de l'orientation .

Les vecteurs $j_{l,i}$ et $j_{A,i}$ ($i = 1, 2, \dots, n$) sont 3×1 . Le vecteur $j_{l,1}$ multiplié à la vitesse articulaire \dot{q} , représente la contribution de l'articulation i à la vitesse linéaire de l'outil. De même, le vecteur $j_{A,i}$, multipliée à la vitesse articulaire \dot{q} représente la contribution de l'articulation i à la vitesse angulaire de l'outil. C'est grâce à ces deux observations que nous pourrons trouver des formules pour ces vecteurs. Cependant, nous devons considérer les articulations rotoïde et prismatiques séparément [35].

II.3.2.2 Modèle Cinématique inverse :

Le modèle cinématique inverse (MCI) donne les vitesses articulaires \dot{q} correspondant aux vitesses souhaitées \dot{X} des organes terminaux. Le modèle cinématique inverse est obtenu en résolvant analytiquement ou numériquement un système d'équations linéaires. La solution analytique réduit considérablement le nombre d'opérations par rapport à la solution numérique, mais il est nécessaire de traiter les cas singuliers séparément. Les solutions numériques sont plus générales Et traitez toutes les situations de la même manière.

➤ Forme générale du modèle cinématique :

Soit $X = [X_p^T \ X_r^T]^T$ une représentation quelconque dans le repère R_0 de la situation du repère R_n fixé à un solide, les éléments X_p et X_r désignant respectivement la position et l'orientation opérationnelles du solide. Les relations entre les vitesses \dot{X}_p et \dot{X}_r et les vecteurs vitesses 0V_n et 0w_n du repère R_n sont telles que :

$$\begin{bmatrix} \dot{X}_p \\ \dot{X}_r \end{bmatrix} = \begin{bmatrix} \Omega_p & \mathbf{0}_3 \\ \mathbf{0}_3 & \Omega_r \end{bmatrix} \begin{bmatrix} V_n^0 \\ w_n^0 \end{bmatrix} = \Omega \begin{bmatrix} V_n^0 \\ w_n^0 \end{bmatrix} \quad (\text{II.22})$$

Les matrices Ω_p et Ω_r dépendant de la représentation choisie respectivement pour la position et pour l'orientation.

A partir de L'équation (II.17) La méthode la plus générale consiste à calculer J^{-1} la matrice inverse de J , qui permet de déterminer les vitesses articulaires \dot{q} grâce à la relation :

$$\dot{q} = J^{-1} \dot{X}$$

II.3.3 Modélisation dynamique :

II.3.3.1 Modèle dynamique direct (MDD) :

Il exprime les accélérations articulaires en fonction des positions, des vitesses et des couples/forces des articulations [43] [44].

Il est représenté par la relation suivante :

$$\ddot{\mathbf{q}} = \mathbf{g}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{\Gamma}, \mathbf{f}_e) \quad (\text{II.23})$$

Avec :

- $\mathbf{\Gamma}$: Vecteur des couples/forces des actionneurs, selon que l'articulation est rotoïde ou prismatique. Dans la suite, on écrira tout simplement couples;
- \mathbf{q} : Vecteur des positions articulaires ;
- $\dot{\mathbf{q}}$: Vecteur des vitesses articulaires ;
- $\ddot{\mathbf{q}}$: Vecteur des accélérations articulaires ;
- \mathbf{f}_e : Vecteur représentant l'effort extérieur (forces et moments) qu'exerce le robot sur l'environnement;

II.3.3.2 Modèle dynamique inverse (MDI) :

Il exprime les couples exercés par les actionneurs en fonction des positions, des vitesses et des accélérations des articulations.

Ce modèle se révèle précieux pour la commande et la planification des trajectoires, car il permet le calcul des efforts articulaires requis afin de suivre une trajectoire spécifiée. Le MDI peut également être employé pour formuler un modèle descripteur adapté à certaines lois de commande.

On représente le modèle dynamique inverse par une relation de la forme [43] :

$$\mathbf{\Gamma} = \mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \mathbf{f}_e) \quad (\text{II.24})$$

➤ Les formalismes les plus utilisés pour le calcul du modèle dynamique inverse sont :

- Formalisme de Lagrange.
- Formalisme de Newton-Euler

II.3.3.3 Formalisme de Newton-Euler :

Ce formalisme se prête admirablement bien au calcul du modèle dynamique (en forme directe ou inverse) en utilisant des coordonnées relatives. Les algorithmes de génération du MDI, tout

Chapitre II : État de l'art sur les commandes et les modélisations appliquées aux robots manipulateur

Comme du MDD, emploient une double récursion (de la base vers l'effecteur et inversement) ; l'expression du torseur dynamique dans un repère associé à une liaison du solide évite la mise en évidence des efforts de réaction dans les liaisons. L'application via un calcul symbolique itératif avec le paramétrage de Denavit-Hartenberg modifié et les paramètres inertiels de base conduit à une obtention systématique.

- La base de ce formalisme est d'appliquer l'équation de Newton et l'équation d'Euler comme suit :

a. Equation de Newton:

$$f = m \dot{v}_c \quad (\text{II.25})$$

b. Equation d'Euler :

$$\mu = I_c \dot{\omega} + \omega \times I_c \omega \quad (\text{II.26})$$

f, μ : sont les vecteurs des forces (ou couples) agissant sur les centre de gravité .

I_c : Est le moment d'inertie d'un corps.

\times : représente l'opération d'un produit vectoriel.

II.3.3.4 Formalisme de Lagrange :

L'approche de modélisation dynamique des mécanismes par la méthode de Lagrange est basée sur le calcul des énergies cinétiques et potentielles et sur la dissipation de ces différentes énergies pour la réalisation du mouvement. La formulation lagrangienne décrit le comportement d'un système dynamique en termes de travail et d'énergie stockée dans le système plutôt que par la force et les moments des membres individuels impliqués [45].

➤ La formulation de Lagrange décrit le comportement d'un système dynamique en terme d'énergie. Les équations de Lagrange sont généralement écrites sous la forme :

$$\Gamma = \frac{d}{dt} \left[\frac{\partial L}{\partial \dot{q}} \right] - \frac{\partial L}{\partial q} \quad (\text{II.27})$$

Où :

L : est le Lagrangien du système, se définit comme la différence entre l'énergie cinétique K et l'énergie potentiel P :

$$L = K - P \quad (\text{II.28})$$

Chapitre II : État de l'art sur les commandes et les modélisations appliquées aux robots manipulateur

q : le vecteur des coordonnées généralisées,

\dot{q} : vecteur de la 1^{ère} dérivée par rapport au temps du vecteur des coordonnées généralisées

F : vecteur des forces généralisées (ou les couples) appliquées au système.

K : l'énergie cinétique.

P : l'énergie potentiel.

Pour obtenir l'équation dynamique générale du robot, on détermine les énergies cinétiques et potentielles, et le lagrangien, puis les remplace dans l'équation de Lagrange.

II.4 Commande Des Robots Manipulateurs

Le problème de la commande d'un robot manipulateur consiste à déterminer comment les actionneurs du robot doivent appliquer des forces généralisées (forces ou couples) pour accomplir une tâche tout en respectant des critères de performance spécifiques. Différentes techniques de commande sont utilisées en fonction de la structure mécanique du bras manipulateur, qui peut influencer le choix du schéma de commande.

Un robot manipulateur est une structure mécanique complexe, et ses inerties varient en fonction de divers facteurs tels que la charge, la configuration, les vitesses et les accélérations. Il existe deux types principaux de mouvements lorsqu'il s'agit de la commande d'un bras manipulateur :

❖ Le premier type de mouvement concerne les mouvements exécutés dans l'espace libre, où le robot peut se déplacer sans contraintes majeures.

❖ Le deuxième type de mouvement implique des mouvements spécifiques dans un espace contraint, généralement avec des forces de contact pour l'organe terminal du robot. Cela se produit lorsque le robot doit interagir avec son environnement.

La plupart des tâches en robotique sont réalisées en combinant ces deux types de mouvements. Pour simplifier la conception de la commande, ces deux types de mouvements sont souvent traités séparément [46].

II.4.1 Commande prédictive:

Ce type de commande utilise un modèle de système et consignes permettant de prédire son évolution et d'agir en fonction des erreurs de prédiction. Trois schémas différents sont proposés : point final fixe, horizon fini et une combinaison des deux premiers schémas. Un

Chapitre II : État de l'art sur les commandes et les modélisations appliquées aux robots manipulateur

avantage majeur de cette approche est que l'erreur de prévision n'est pas affectée par le bruit de mesure, mais la dépendance au modèle reste forte [47].

II.4.2 Commande adaptative :

Proposition d'un contrôle adaptatif non linéaire asymptotiquement stable. Étant donné que le modèle dynamique d'un robot ne peut pas être parfait, il est nécessaire que les ingénieurs appliquent des théories de contrôle adaptatif pour estimer ou ajuster les valeurs des paramètres dynamiques utilisés dans le contrôle en ligne. Utilisation des propriétés de passivité pour développer une loi de commande adaptative non linéaire asymptotiquement stable. Le contrôle adaptatif est un système adaptatif qui mesure un indice de performance (IP) en utilisant les entrées et sorties d'un système réglable. Cette performance est comparée à un ensemble de valeurs de référence. En fonction de cette comparaison, le mécanisme d'adaptation modifie les paramètres du système et génère des commandes supplémentaires pour maintenir la performance du système à proximité des valeurs de référence.

Il existe cinq stratégies significatives dans le domaine de la commande adaptative pour les robots manipulateurs:

1. Simplification du modèle dynamique.
2. Application des techniques adaptatives développées pour les systèmes linéaires au cas des robots.
3. Proposition d'une commande adaptative non linéaire asymptotiquement stable.
4. Utilisation des propriétés de la passivité pour développer une loi de commande adaptative non linéaire asymptotiquement stable.
5. Utilisation pour l'adaptation d'un modèle qui n'est pas fonction des accélérations articulaires tel que le modèle dynamique filtré ou le modèle énergétique [46].

➤ Le schéma général d'un système de commande adaptatif peut être le suivant :

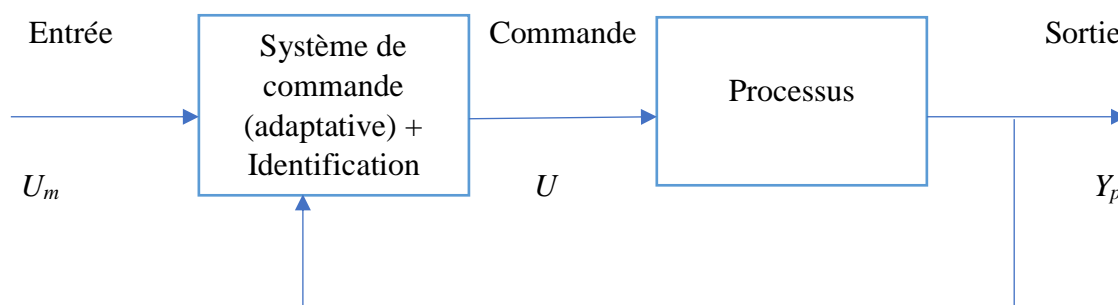


Figure II.16 : Schéma général de la commande adaptative [46].

Chapitre II : État de l'art sur les commandes et les modélisations appliquées aux robots manipulateur

Ce type de techniques vise à corriger les déficiences de la commande par découplage non linéaire, comme la connaissance approximative des paramètres du modèle du robot ou pour s'adapter aux différentes conditions opératoires. Ce type de schémas cherche à estimer ou ajuster en ligne les valeurs des paramètres utilisées dans le calcul de la loi de commande. Un travail plus intéressant sur ce sujet est la commande proposée par la commande de Slotine-Li ou commande adaptative passive. Les avantages de ce type de techniques sont évidents, malheureusement la puissance de calcul demandée au système constitue un inconvénient important.

II.4.2.1 Limitations de la Commande Adaptative :

Le problème de la conception de lois de commande adaptatives pour les robots manipulateurs est d'assurer le suivi de trajectoire souhaité. Pour compenser cette incertitude paramétrique, de nombreux chercheurs ont proposé des stratégies de contrôle adaptatif. L'un des avantages de l'approche adaptative est que la précision du manipulateur lorsqu'il s'agit de transporter des charges inconnues s'améliore avec le temps. Le développement de contrôleurs adaptatifs efficaces représente une étape importante vers des applications robotiques à grande vitesse et précision. Même dans une installation industrielle bien structurée, les robots peuvent être confrontés à des incertitudes concernant les paramètres décrivant les propriétés dynamiques de la charge de portée.

L'utilisation des techniques adaptatives est motivée par l'ajustement automatique des paramètres du système quand ils sont inconnus et ont des variations imprévisibles dans le temps. Les contrôleurs adaptatifs sont formulés en séparant les paramètres constants inconnus des fonctions connues dans l'équation dynamique de robot.

II.4.3 Commande par articulation :

Cette technique est utilisée par des robots manipulateurs qui utilisent des servomoteurs avec de forts rapports de réduction. Lorsque le système présente un comportement linéaire, l'asservissement du mouvement peut être réalisé par des techniques classiques de commande. Nous parlons alors d'une commande décentralisée de type PID. le schéma classique est amélioré avec des signaux d'anticipation pour corriger les effets de la force de gravité et de couplage [48].

II.4.4 Commande jacobienne :

Cette méthode est en utilisation depuis 1969, et elle est communément désignée par le nom de "commande à mouvement résolu" lorsqu'elle repose sur l'utilisation de la matrice jacobienne

Chapitre II : État de l'art sur les commandes et les modélisations appliquées aux robots manipulateur

Inverse du bras manipulateur pour calculer les vitesses de consigne aux articulations. Une autre appellation courante pour cette approche est "commande résolue au niveau cinématique". Les approches les plus fréquentes de cette méthode incluent : Mouvement à vitesse résolu, Mouvement à accélération résolu et Mouvement à force résolu. La technique de mouvement résolu commande la position de l'organe terminal du manipulateur dans l'espace cartésien, par combinaison des mouvements de plusieurs articulations.

Ce type de schémas peut se présenter sous la forme d'un schéma cinématique quand les vitesses calculées aux articulations sont utilisées directement comme des consignes pour les boucles de commande de chaque articulation, ou sous la forme d'un schéma dynamique quand le schéma utilise le modèle dynamique pour découpler les articulations. Son principal inconvénient est l'utilisation de l'inverse de la matrice jacobienne, qui peut devenir singulière dans certaines conditions [48].

II.4.5 Commande passive :

Cette technique considère le robot comme un système passif, c'est à dire un système qui dissipe de l'énergie. De telles lois de commande permettent de modifier l'énergie naturelle du robot pour qu'il réalise la tâche. En utilisant le formalisme de Hamilton, la commande cherche à minimiser l'énergie du système en utilisant un bloc non linéaire passif dans la boucle de retour. La commande passive tend à être plus robuste que le découplage non linéaire, lorsque la technique ne recherche pas à annuler les non linéarités [46].

• Les Avantages et Limitations de la commande passive :

➤ Avantages:

- Robustesse : La commande passive est souvent plus robuste face aux incertitudes et aux variations dans le modèle du système. Elle peut fonctionner efficacement même en présence de perturbations imprévues.
- Consommation Énergétique Réduite : Étant donné que la commande passive exploite les propriétés naturelles du système, elle peut réduire la consommation d'énergie par rapport à d'autres approches de commande plus énergivores.
- Simplicité de Conception : Les lois de commande passives peuvent être plus simples à concevoir et à mettre en œuvre par rapport à d'autres méthodes de commande complexe.

Chapitre II : État de l'art sur les commandes et les modélisations appliquées aux robots manipulateur

➤ Inconvénients :

- Limitation de Performance : La commande passive peut ne pas être la meilleure option pour des tâches nécessitant une précision extrême ou une dynamique de mouvement rapide.
- Complexité des Modèles : Dans certains cas, il peut être complexe de modéliser et de concevoir des systèmes de commande passive, en particulier pour des systèmes hautement non linéaires.

II.4.6 Commande optimale :

Pour réaliser une tâche, il peut exister un grand nombre de solutions. Dans ce cas, il peut être souhaitable de choisir une solution qui minimise un certain critère. La littérature présente différents types de critères pour la commande optimale : la commande en temps minimal, du domaine des neurosciences la minimisation du jerk pour maximiser la souplesse du mouvement, entre autres [49].

II.4.7 Commande robuste :

Avec des paramètres fixes, les techniques de découplage non linéaire sont connues pour devenir instables en présence d'incertitude. Si les paramètres du modèle sont inconnus et que l'incertitude des paramètres permet une plage connue, Des techniques de contrôle robustes peuvent alors être utilisées. Par exemple, les travaux de Slotine ont examiné les techniques de mode glissant, également connues sous le nom de contrôle de structure variable. Cette technique utilise des surfaces de glissement où la stabilité du système est garantie [49].

• Les Avantages et Limitations de la commande robuste :

La commande robuste présente des avantages significatifs, notamment une grande stabilité en présence de perturbations et d'incertitudes, une tolérance aux variations de modèle, une fiabilité accrue et une réduction des risques liés à l'instabilité. Cependant, elle comporte également des inconvénients tels que sa complexité de conception, son conservatisme potentiel qui limite les performances dans des conditions nominales, la difficulté de mise au point des paramètres, et un coût potentiellement élevé en termes de temps et de ressources.

II.5 Conclusion :

Ce chapitre découvre quelques concepts fondamentaux dans le domaine de la robotique, notamment dans les sections I.1 et I.2, où sont introduits les définitions et concepts généraux de base, permettant de comprendre la composition et le fonctionnement des robots manipulateurs.

La section II.1 est consacrée à la modélisation des manipulateurs robotiques, depuis la modélisation géométrique, la modélisation cinématique, jusqu'à la modélisation dynamique représentant la forme lagrangienne des robots manipulateurs. La dernière section explique les avantages et les inconvénients du contrôle adaptatif et présente d'autres méthodes de commande.

En résumé, les approches de modélisation et de commande évoluent rapidement grâce aux avancées technologiques, notamment l'intégration de l'IA. Les défis futurs incluent la gestion de la complexité, l'assurance de la sécurité des systèmes autonomes et la résolution de problèmes dans des environnements dynamiques et incertains.

Dans le chapitre suivant, l'optimisation des trajectoires des robots manipulateurs sera consacrée à l'optimisation des trajectoires avec 2ddl.

Chapitre III

Recherche sur l'optimisation de la trajectoire d'un robot manipulateur à 2 DDL

III.1 Introduction

Dans ce troisième chapitre de notre recherche sur l'optimisation de la trajectoire des robots manipulateurs à deux degrés de liberté (DDL), nous abordons un domaine essentiel de la robotique moderne. Les robots à 2DDL, polyvalents et largement utilisés dans divers secteurs, nécessitent une planification de trajectoire optimale pour maximiser leur efficacité, précision et sécurité.

L'objectif principal de ce chapitre est de plonger profondément dans la complexité de l'optimisation de trajectoire pour ces robots spécifiques. Nous visons à comprendre les défis liés à leurs contraintes cinématiques et dynamiques, tout en examinant les méthodes d'optimisation disponibles. Nous mettons en lumière les avantages et les limites de ces approches, étayant nos arguments par des études de cas concrets et des résultats empiriques.

Enfin, nous discutons des implications de nos découvertes pour diverses applications robotiques, soulignant l'impact potentiel sur l'efficacité, la précision et la sécurité des robots à 2DDL. Ce chapitre ouvre la voie à des avancées significatives dans ce domaine crucial de la robotique, promettant des améliorations pratiques dans de nombreuses applications.

III.2 Modélisation du robot manipulateur à deux dll

Dans le chapitre précédent, nous avons exploré en détail la modélisation géométrique, cinématique et dynamique des robots manipulateurs. Dans ce chapitre, nous allons nous concentrer sur l'application de ces concepts de modélisation à un robot manipulateur spécifique à 2 degrés de liberté (DDL). Alors que les chapitres précédents nous ont fourni les bases théoriques nécessaires, nous allons maintenant les mettre en pratique pour résoudre des problèmes concrets liés à la trajectoire et au mouvement de ce robot.

Dans cette section, nous nous concentrons sur l'application pratique des concepts de modélisation dynamique aux robots manipulateurs à 2 degrés de liberté (DDL).

III.2.1 Description du robot manipulateur :

Le robot manipulateur à 2 degrés de liberté étudié dans cette recherche est un système mécanique composé de deux bras articulés et contrôlés, permettant des mouvements dans un plan bidimensionnel. Voici une description détaillée de ses caractéristiques physiques et de son contexte d'application :

Chapitre III : Recherche sur l'optimisation de la trajectoire d'un robot manipulateur à 2 DDI

- **Structure mécanique :** Ce robot manipulateur se compose de deux bras articulés. Le premier bras, souvent appelé "bras de base", est monté sur une base fixe. Le second bras, également appelé "bras d'extrémité", est attaché à l'extrémité du premier bras. Les deux bras sont reliés par un joint rotatif à un degré de liberté. Le premier bras est monté sur un second joint rotatif qui le relie à la base fixe. Cette configuration permet au robot de réaliser des mouvements de rotation autour de deux axes
- **Longueur des bras :** La longueur des bras peut varier en fonction du modèle spécifique du robot manipulateur, mais elle est généralement définie en fonction des besoins de l'application. Une description précise de la longueur des bras spécifiques utilisés dans votre étude doit être incluse.
- **Joints :** Le robot dispose de deux types de joints principaux : les joints rotatifs. Chacun de ces joints permet au robot de réaliser un mouvement angulaire. Les valeurs des angles des joints déterminent la configuration du robot à un moment donné.
- **Capteurs :** Pour surveiller et contrôler le robot, différents types de capteurs peuvent être intégrés. Cela peut inclure des capteurs de position pour mesurer les angles des joints, des capteurs de force et de pression pour détecter les contacts avec l'environnement, ainsi que des capteurs de vision pour la perception visuelle.

III.2.1.1 Domaine d'application :

Le robot manipulateur à 2 degrés de liberté trouve des applications dans divers domaines, notamment l'industrie manufacturière, la recherche médicale, l'automatisation des laboratoires, et bien d'autres. Par exemple, dans l'industrie, il peut être utilisé pour des opérations de manipulation simples telles que le tri, le chargement, le déchargement ou l'assemblage de pièces

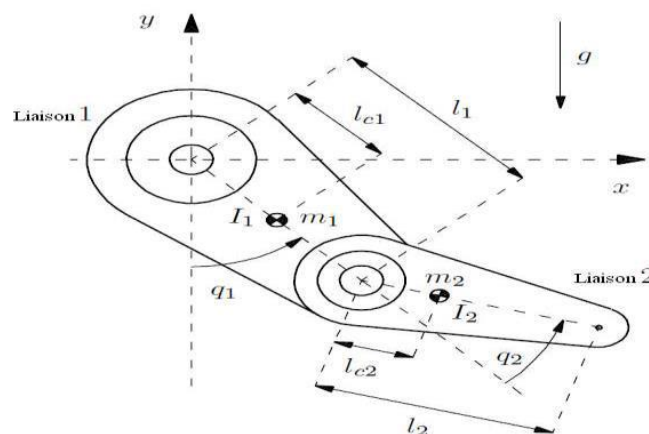


Figure III.1: Diagramme du bras pelican à deux degré de liberté (2ddl) [50].

Chapitre III : Recherche sur l'optimisation de la trajectoire d'un robot manipulateur à 2 DDI

- Considérons le robot à deux articulations représenté sur la figure (III.1) avec les spécifications suivantes en coordonnées Oxy [50]:

Paramètre	Valeur
- Masse de la liaison 1 (m1)	6.5225 kg
- Masse de la liaison 2 (m2)	2.0458 kg
- Longueur de la liaison 1 (l1)	0.26 m
- Longueur de la liaison 2 (l2)	0.26 m
- Distance vers le centre de masse de la liaison 1(lc1)	0.0983 m
- Distance vers le centre de masse de la liaison 2(lc2)	0.0229 m
- Moment d'inertie du centre de masse m1 (I1)	0.1213 kg m ²
- Moment d'inertie du centre de masse m2 (I2)	0.0116 kg m ²
- La gravité (g)	9.81 ms ⁻²

Tableau III.1: paramètres du robot (pelican).

III.2.2 Modélisation dynamique :

Dans cette section nous allons écrire le modèle dynamique du bras en utilisant la formulation Lagrange-Euler (LE) :

$$\Gamma = \frac{d}{dt} \left[\frac{\partial L}{\partial \dot{q}} \right] - \frac{\partial L}{\partial q} \quad (\text{III. 1})$$

○ Considérons le robot décrit dans la figure (III.1) L'énergie cinétique K du bras est décrit par la somme :

$$K = \sum_{i=1}^n k_i$$
$$K(\mathbf{q}, \dot{\mathbf{q}}) = k_1(\mathbf{q}, \dot{\mathbf{q}}) + k_2(\mathbf{q}, \dot{\mathbf{q}}) \quad (\text{III.2})$$

Avec :

k_1 et k_2 sont les énergies cinétiques associées au centres de masses m_1 et m_2 respectivement

Chapitre III : Recherche sur l'optimisation de la trajectoire d'un robot manipulateur à 2 DDI

Dans notre travail, nous avons considéré d'abord un robot manipulateur à 2ddl, donné par figure (III.1), dont le mouvement des articulations est rotoïde, nous avons donc $q_i = \theta_i$

Tel que :

$$q = [\theta_1, \theta_2]^T \quad \tau = [\tau_1, \tau_2]^T$$

➤ Les coordonnées du centre de masse de la liaison 1 selon le plant (xy) :

$$x_1 = l_{c1} \sin(q_1) \quad (III.3)$$

$$y_1 = -l_{c1} \cos(q_1) \quad (III.4)$$

Le vecteur de vitesse v_1 de la même liaison est :

$$v_1 = \begin{bmatrix} \dot{x}_1 \\ \dot{y}_1 \end{bmatrix} = \begin{bmatrix} l_{c1} \cos(q_1) \dot{q}_1 \\ l_{c1} \sin(q_1) \dot{q}_1 \end{bmatrix} \quad (III.5)$$

Sachant que :

$$\|v_1\|^2 = v_1^T v_1 = l_{c1}^2 \dot{q}_1^2 \quad (III.6)$$

L'énergie cinétique correspondante à la 1^{ère} liaison est obtenu par :

$$k_1(q, \dot{q}) = \frac{1}{2} m_1 v_1^T v_1 + \frac{1}{2} I_1 \dot{q}_1^2 = \frac{1}{2} m_1 l_{c1}^2 \dot{q}_1^2 + \frac{1}{2} I_1 \dot{q}_1^2 \quad (III.7)$$

➤ Les coordonnées du centre de masse de la liaison 2 suivant le plant (xy) :

$$x_2 = l_1 \sin(q_1) + l_{c2} \sin(q_1 + q_2) \quad (III.8)$$

$$y_2 = -l_1 \cos(q_1) - l_{c2} \cos(q_1 + q_2) \quad (III.9)$$

Le vecteur de vitesse v_2 de la même liaison (2) est :

$$v_2 = \begin{bmatrix} \dot{x}_2 \\ \dot{y}_2 \end{bmatrix} = \begin{bmatrix} l_1 \cos(q_1) \dot{q}_1 + l_{c2} \cos(q_1 + q_2) [\dot{q}_1 + \dot{q}_2] \\ l_1 \sin(q_1) \dot{q}_1 + l_{c2} \sin(q_1 + q_2) [\dot{q}_1 + \dot{q}_2] \end{bmatrix} \quad (III.10)$$

- En utilisant les deux formules trigonométriques :

$$\cos(\theta)^2 + \sin(\theta)^2 = 1 \quad \text{Et} \quad \sin(q_1) \sin(q_1 + q_2) + \cos(q_1) \cos(q_1 + q_2) = \cos(q_2)$$

Ont conclu que :

Chapitre III : Recherche sur l'optimisation de la trajectoire d'un robot manipulateur à 2 DDI

$$\begin{aligned} \|v_2\|^2 = v_2^T v_2 = l_1^2 \dot{q}_1^2 + l_{c2}^2 [\dot{q}_1^2 + 2\dot{q}_1 \dot{q}_2 + \dot{q}_2^2] \\ + 2l_1 l_{c2} [\dot{q}_1^2 + \dot{q}_1 \dot{q}_2] \cos(q_2) \end{aligned} \quad (\text{III.11})$$

L'énergie cinétique correspondante à la 2^{ème} liaison est obtenu par :

$$k_2(q, \dot{q}) = \frac{1}{2} m_2 v_2^T v_2 + \frac{1}{2} I_2 (\dot{q}_1 + \dot{q}_2)^2 \quad (\text{III.12})$$

On remplace l'équation (III. 13) dans l'équation (III. 14) :

$$\begin{aligned} k_2(q, \dot{q}) = \frac{1}{2} m_2 l_1^2 \dot{q}_1^2 + \frac{1}{2} m_2 l_{c2}^2 [\dot{q}_1^2 + 2\dot{q}_1 \dot{q}_2 + \dot{q}_2^2] + m_2 l_1 l_{c2} [\dot{q}_1^2 + \\ \dot{q}_1 \dot{q}_2] \cos(q_2) + \frac{1}{2} I_2 (\dot{q}_1 + \dot{q}_2)^2 \end{aligned} \quad (\text{III.15})$$

○ L'énergie potentielle totale est la somme de l'énergie potentielle dans les différentes articulations, c'est-à-dire :

$$P = \sum_{i=1}^n p_i \quad (\text{III.16})$$

$$P = p_1(q) + p_2(q) \quad (\text{III.17})$$

Avec p_1 et p_2 sont les énergies potentiels associées aux masses m_1 et m_2 respectivement.

Si l'énergie potentiel est égale à zéro lorsque $y=0$ alors :

$$p_1(q) = -m_1 l_{c1} g \cos(q_1) \quad (\text{III.18})$$

$$p_2(q) = -m_2 l_1 g \cos(q_1) - m_2 l_{c2} g \cos(q_1 + q_2) \quad (\text{III.19})$$

➤ Détermination du Lagrangien $L = K - P$

$$\mathcal{L}(q, \dot{q}) = [k_1(q, \dot{q}) + k_2(q, \dot{q})] - [p_1(q) + p_2(q)] \quad (\text{III.20})$$

$$\begin{aligned} \mathcal{L}(q, \dot{q}) = \frac{1}{2} m_1 l_{c1}^2 \dot{q}_1^2 + \frac{1}{2} I_1 \dot{q}_1^2 + \frac{1}{2} m_2 l_1^2 \dot{q}_1^2 + \frac{1}{2} m_2 l_{c2}^2 [\dot{q}_1^2 + 2\dot{q}_1 \dot{q}_2 + \dot{q}_2^2] + \\ m_2 l_1 l_{c2} [\dot{q}_1^2 + \dot{q}_1 \dot{q}_2] \cos(q_2) + \frac{1}{2} I_2 (\dot{q}_1 + \dot{q}_2)^2 + m_1 l_{c1} g \cos(q_1) + \\ m_2 l_1 g \cos(q_1) + m_2 l_{c2} g \cos(q_1 + q_2) \end{aligned} \quad (\text{III.21})$$

Chapitre III : Recherche sur l'optimisation de la trajectoire d'un robot manipulateur à 2 DDI

$$\begin{aligned} \mathcal{L}(q, \dot{q}) = & \frac{1}{2} [m_1 l_{c1}^2 + m_2 l_1^2] \dot{q}_1^2 + \frac{1}{2} m_2 l_{c2}^2 [\dot{q}_1^2 + 2\dot{q}_1 \dot{q}_2 + \dot{q}_2^2] + \\ & m_2 l_1 l_{c2} [\dot{q}_1^2 + \dot{q}_1 \dot{q}_2] \cos(q_2) + [m_1 l_{c1} + m_2 l_1] g \cos(q_1) + \\ & m_2 g l_{c2} \cos(q_1 + q_2) + \frac{1}{2} I_1 \dot{q}_1^2 + \frac{1}{2} I_2 (\dot{q}_1 + \dot{q}_2)^2 \end{aligned} \quad (\text{III.22})$$

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \dot{q}_1} = & [m_1 l_{c1}^2 + m_2 l_1^2] \dot{q}_1 + m_2 l_{c2}^2 \dot{q}_1 + m_2 l_{c2}^2 \dot{q}_1 + m_2 l_{c2}^2 \dot{q}_1 + m_2 l_{c2}^2 \dot{q}_2 + \\ & 2m_2 l_1 l_{c2} \cos(q_2) \dot{q}_1 + m_2 l_1 l_{c2} \cos(q_2) \dot{q}_2 + I_1 \dot{q}_1 + I_2 [\dot{q}_1 + \dot{q}_2] \end{aligned} \quad (\text{III.23})$$

$$\begin{aligned} \frac{d}{dt} \left[\frac{\partial \mathcal{L}}{\partial \dot{q}_1} \right] = & [m_1 l_{c1}^2 + m_2 l_1^2 + m_2 l_{c2}^2 + 2m_2 l_1 l_{c2} \cos(q_2)] \ddot{q}_1 + [m_2 l_{c2}^2 + m_2 l_1 l_{c2} \cos(q_2)] \ddot{q}_2 \\ & - 2m_2 l_1 l_{c2} \sin(q_2) \dot{q}_1 \dot{q}_2 - m_2 l_1 l_{c2} \sin(q_2) \dot{q}_2^2 + I_1 \ddot{q}_1 + I_2 [\ddot{q}_1 + \ddot{q}_2] \end{aligned} \quad (\text{III.24})$$

$$\frac{\partial \mathcal{L}}{\partial q_1} = -[m_1 l_{c1} + m_2 l_1] g \sin(q_1) - m_2 g l_{c2} \sin(q_1 + q_2)$$

$$\frac{\partial \mathcal{L}}{\partial \dot{q}_2} = m_2 l_{c2}^2 \dot{q}_1 + m_2 l_{c2}^2 \dot{q}_2 + m_2 l_1 l_{c2} \cos(q_2) \dot{q}_1 + I_2 [\dot{q}_1 + \dot{q}_2] \quad (\text{III.24})$$

$$\begin{aligned} \frac{d}{dt} \left[\frac{\partial \mathcal{L}}{\partial \dot{q}_2} \right] = & m_2 l_{c2}^2 \ddot{q}_1 + m_2 l_{c2}^2 \ddot{q}_2 + m_2 l_1 l_{c2} \cos(q_2) \ddot{q}_1 - m_2 l_1 l_{c2} \sin(q_2) \dot{q}_1 \dot{q}_2 + \\ & I_2 [\ddot{q}_1 + \ddot{q}_2] \end{aligned} \quad (\text{III.25})$$

$$\frac{\partial \mathcal{L}}{\partial q_2} = -m_2 l_1 l_{c2} \sin(q_2) [\dot{q}_1 \dot{q}_2 + \dot{q}_1^2] - m_2 g l_{c2} \sin(q_1 + q_2) \quad (\text{III.26})$$

➤ Le modèle dynamique du bras est obtenu en appliquant l'équation de Lagrange suivante :

$$\frac{d}{dt} \left[\frac{\partial \mathcal{L}}{\partial \dot{q}_i} \right] - \frac{\partial \mathcal{L}}{\partial q_i} = \tau_i \quad \text{Avec : } i=1,2$$

Chapitre III : Recherche sur l'optimisation de la trajectoire d'un robot manipulateur à 2 DDI

➤ On obtient le modèle dynamique au final :

$$\begin{aligned} \tau_1 = & [m_1 l_{c1}^2 + m_2 l_1^2 + m_2 l_{c2}^2 + 2m_2 l_1 l_{c2} \cos(q_2) I_1 + I_2] \ddot{q}_1 + [m_2 l_{c2}^2 + \\ & m_2 l_1 l_{c2} \cos(q_2) + I_2] \ddot{q}_2 - 2m_2 l_1 l_{c2} \sin(q_2) \dot{q}_1 \dot{q}_2 - m_2 l_1 l_{c2} \sin(q_2) \dot{q}_2^2 + \\ & [m_1 l_{c1} + m_2 l_1] g \sin(q_1) + m_2 g l_{c2} \sin(q_1 + q_2) \end{aligned} \quad (III.27)$$

$$\begin{aligned} \tau_2 = & [m_2 l_{c2}^2 + m_2 l_1 l_{c2} \cos(q_2) + I_2] \ddot{q}_1 + [m_2 l_{c2}^2 + I_2] \ddot{q}_2 + m_2 l_1 l_{c2} \sin(q_2) \dot{q}_2^2 + \\ & m_2 g l_{c2} \sin(q_1 + q_2) \end{aligned} \quad (III.28)$$

○ Sachant τ_1 et τ_2 que sont des couples délivrés aux actionneurs des deux articulation 1 et 2 [50].

En robotique, si le mouvement des articulations constituant le robot manipulateur est rotoïde, l'expression du couple peut être s'exprimer sous la forme matricielle suivante, ce qui correspondant au modèle dynamique du robot : :

$$\underbrace{\begin{bmatrix} M_{11}(q) & M_{12}(q) \\ M_{21}(q) & M_{22}(q) \end{bmatrix}}_{M(q)} \ddot{q} + \underbrace{\begin{bmatrix} C_{11}(q, \dot{q}) & C_{12}(q, \dot{q}) \\ C_{21}(q, \dot{q}) & C_{22}(q, \dot{q}) \end{bmatrix}}_{C(q, \dot{q})} \dot{q} + \underbrace{\begin{bmatrix} g_1(q) \\ g_2(q) \end{bmatrix}}_{g(q)} = \tau \quad (III.29)$$

Avec :

$$M_{11}(q) = m_1 l_{c1}^2 + m_2 [l_1^2 + l_{c2}^2 + 2l_1 l_{c2} \cos(q_2)] + I_1 + I_2$$

$$M_{12}(q) = m_2 [l_1^2 + l_{c2}^2 + 2l_1 l_{c2} \cos(q_2)] + I_2$$

$$M_{21}(q) = m_2 [l_1^2 + l_{c2}^2 + 2l_1 l_{c2} \cos(q_2)] + I_2$$

$$M_{22}(q) = [m_2 l_{c2}^2 + I_2]$$

$$C_{11}(q, \dot{q}) = -m_2 l_1 l_{c2} \sin(q_2) \dot{q}_2$$

$$C_{12}(q, \dot{q}) = -m_2 l_1 l_{c2} \sin(q_2) [\dot{q}_1 + \dot{q}_2]$$

$$C_{21}(q, \dot{q}) = m_2 l_1 l_{c2} \sin(q_2) \dot{q}_1$$

$$C_{22}(q, \dot{q}) = 0$$

$$g_1(q) = [m_1 l_{c1} + m_2 l_1] g \sin(q_1) + m_2 g l_{c2} \sin(q_1 + q_2)$$

$$g_2(q) = m_2 g l_{c2} \sin(q_1 + q_2)$$

Chapitre III : Recherche sur l'optimisation de la trajectoire d'un robot manipulateur à 2 DDI

Tel que :

q : vecteur des variables articulaires généralisées.

\dot{q} : vecteur des vitesses angulaires.

\ddot{q} : vecteur des accélérations angulaires.

M : matrice d'inertie.

C : matrice spécifiant l'effet de centrifuge et de Coriolis.

g : vecteur exprimant l'effet gravitationnel.

τ : vecteur des couples généralisés.

III.3 Notion générale sur l'optimisation de trajectoire

L'optimisation de trajectoire est un domaine essentiel de la robotique et de l'automatisation, visant à trouver la meilleure trajectoire qu'un objet, un véhicule ou un robot devrait suivre pour accomplir une tâche spécifique. Cette tâche peut être aussi simple que déplacer un robot d'un point A à un point B ou aussi complexe que la planification de trajectoires pour des véhicules autonomes dans des environnements en constante évolution. Voici quelques notions générales sur l'optimisation de trajectoire :

III.3.1 Les définitions de la trajectoire et de l'optimisation :

○ Trajectoire : La trajectoire désigne le chemin suivi par un objet, un véhicule ou un robot dans l'espace en fonction du temps. Elle est essentielle pour comprendre comment un objet se déplace d'un point à un autre.

○ Optimisation : est le processus de recherche de la meilleure solution parmi plusieurs possibilités pour un problème donné. Dans le contexte de la planification de trajectoire, l'optimisation vise à trouver la trajectoire la plus efficace, sûre ou appropriée pour un robot, en minimisant ou maximisant une fonction objectif tout en respectant des contraintes spécifiques.

III.3.1.1 Chemins et trajectoires :

La plupart des travaux de planification de trajectoire sont basés sur le concept d'espace des configurations du robot. Une configuration désigne l'ensemble des paramètres caractérisant d'une manière unique le robot dans son environnement ou espace de travail. L'ensemble des configurations du robot est l'espace des configurations Q qui a une structure de variété différentielle. On note n la dimension de Q [51].

Chapitre III : Recherche sur l'optimisation de la trajectoire d'un robot manipulateur à 2 DDI

Soit $x \in R^n$ la position d'un point dans l'espace à n dimensions. Un chemin φ de R^n , ($\varphi \subset R^n$ est un ensemble de positions successives de R^n) relie une position initiale x_0 à une position finale x_f .

Certains chemins peuvent être représentés sous la forme d'une liste de $m + 1$ positions successives, telles que $x_i \in \varphi$ avec $0 \leq i \leq m$. Lorsque le chemin est défini par les seules positions initiale x_0 et finale x_f on parle de mouvement point à point.

Dans la pratique, il arrive souvent que l'on désire conduire un système dynamique d'un état initial x_0 à un état final x_f . C'est ce qu'on appelle un problème de planification de trajectoire. Pour résoudre un tel problème, il faut qu'il existe au moins une fonction d'entrée $u(t)$ produisant une trajectoire du système passant par les états x_0 et x_f .

Définissons la trajectoire TR d'un point comme l'ensemble des positions successives de ce point quand le temps t varie de t_0 à t_f pour un déplacement de x_0 à x_f [52].

$$\begin{aligned} TR: [t_0, t_f] &\rightarrow \varphi \subset R^n \\ t &\rightarrow TR(t) \end{aligned}$$

Un chemin φ est un ensemble de positions de l'espace sans relation temporelle. Une trajectoire TR est une fonction du temps qui définit une loi d'évolution sur un chemin. La trajectoire définit les caractéristiques cinématiques du mouvement.

Une trajectoire est une fonction continue de $[t_{initial}, t_{final}] \subset R$ dans Q qui à toute valeur $t \in [t_{initial}, t_{final}]$ Associe une configuration [3]:

$$\begin{aligned} q: [t_{initial}, t_{final}] &\rightarrow Q \\ t &\rightarrow q(t) \end{aligned}$$

Une trajectoire est dite admissible si elle est solution du système d'équations différentielles correspondant au modèle cinématique du robot, incluant les contraintes sur les commandes et sur les conditions initiales et finales imposées. Un chemin est l'image d'une trajectoire dans Q . Un chemin admissible est l'image d'une trajectoire admissible.

La planification est également une sous-discipline de l'intelligence artificielle (IA) qui

Se propose :

- Etant donnée une représentation de l'état initial de la situation (monde dans le jargon de l'IA).
- Etant donné un ensemble d'opérateurs de changement d'état de la situation (qui représentent les actions qu'il est possible d'effectuer).

Chapitre III : Recherche sur l'optimisation de la trajectoire d'un robot manipulateur à 2 DDI

- Etant donné un but à atteindre (problème à résoudre).

Fournit une méthode permettant à un système informatique de trouver une séquence d'actions (c'est-à-dire une séquence d'opérateurs directement exécutables) qui peut être appliquée à une situation pour la faire passer d'un état initial à un état qui satisfait l'objectif à atteindre. Un plan est un ensemble structuré d'actions pour atteindre un objectif. La planification (ou génération de plans) est également le processus de formulation de cet ensemble d'actions. En créant un plan, un système informatique peut mémoriser la séquence d'opérations pour résoudre un problème donné. Le plan est élaboré par une partie du système appelée générateur de plan (ou planificateur).

III.3.2 Planification des chemins et planification des trajectoires :

En robotique, le terme planification des chemins fait référence à la courbe géométrique qui relie deux points différents. Cependant, la planification des trajectoires décrit cette courbe géométrique en termes de temps. Habituellement, la planification des chemins précède la planification des trajectoires dans le sens que la planification des chemins trouve une courbe réalisable qui relie la position initiale à la cible, tandis que la planification des trajectoires assigne la variable de temps à chaque point le long de cette courbe.

III.3.2.1 Planification des chemins :

La planification de chemin est la construction de formes géométriques que le robot peut suivre sans aucune référence à une échelle de temps. D'une manière générale, la difficulté de cette question dépend de la situation. Dans le cas d'une trajectoire pour un manipulateur, la géométrie est déterminée par la tâche du robot. Cependant, pour les robots mobiles, le problème de planification de trajectoire est plus exigeant. Étant donné que ces types de robots nécessitent plus d'autonomie, la fonctionnalité de planification de chemin a été étendue pour inclure le comportement d'évitement des obstacles. Cependant, le type de planification du trajet est fortement influencé par l'environnement dans lequel le robot navigue. Dans un environnement statique, la planification du chemin s'effectue principalement hors ligne, mais dans un environnement dynamique inconnu, le chemin doit être planifié en ligne [53].

III.3.2.2 Planification des trajectoires:

On appelle planification de trajectoire, le calcul d'une trajectoire admissible et sans collision pour un robot entre une configuration de départ et une configuration d'arrivée données.

Il faut noter que la planification de trajectoire diffère selon que l'on ait à faire à un robot

Chapitre III : Recherche sur l'optimisation de la trajectoire d'un robot manipulateur à 2 DDI

Mobile ou à un bras manipulateur.

Pour les bras manipulateurs, la planification consiste à générer des trajectoires que l'organe terminal (effecteur) doit effectuer tout en tenant compte de :

- l'admissibilité de la trajectoire,
- de l'espace d'atteignable,
- des contraintes sur la vitesse d'exécution,
- des contraintes sur les capacités "physiques" du bras manipulateur quant à la tâche à effectuer, ...etc.

III.3.2.3 Génération de la trajectoire dans l'espace articulaire :

La génération de la trajectoire dans l'espace articulaire donne en résultat un ensemble de données : position, vitesse et accélération articulaires ($\theta, \dot{\theta}, \ddot{\theta}$) qui sont utilisées comme un signal de référence. Pour une position initiale et finale données dans l'espace de travail on utilise la géométrie inverse pour déterminer les angles articulaires correspondant à cette position et même pour l'orientation, la position initiale du manipulateur devient un ensemble d'angles articulaires de départ et la position finale correspond à un autre ensemble d'angles

articulaires d'arrivée, ce qui est demandé pour la planification de la trajectoire est de trouver une fonction lisse pour chaque articulation dont la valeur à l'instant t_0 est la position initiale de l'articulation et dont la valeur à l'instant t_f est la position désirée de cette même articulation. Il y a beaucoup de fonctions lisses $\theta(t)$ qui pourraient être utilisées pour interpoler les valeurs d'angles articulaires [54].

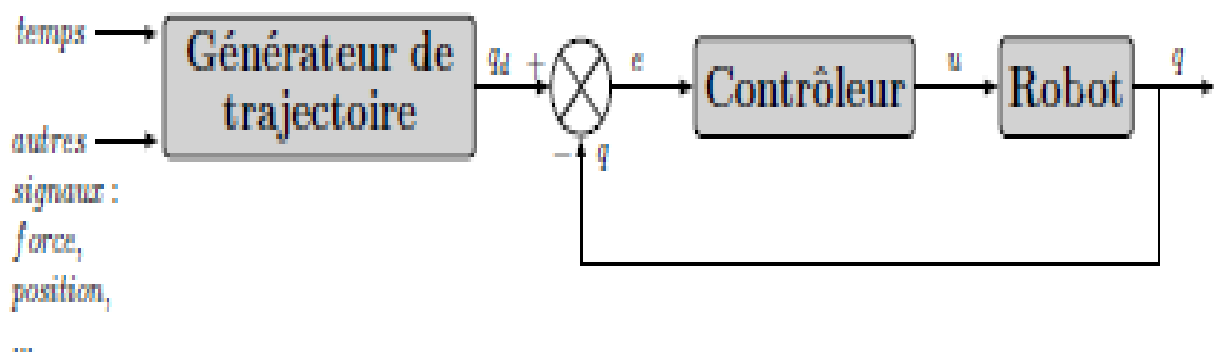


Figure III.2 : Planification dans l'espace articulaire [31].

III.3.3 Génération de trajectoire en environnement connu :

La génération de trajectoire consiste à interpréter toutes les informations disponibles afin de spécifier une séquence d'actions à imposer au robot. Ces actions peuvent s'exprimer sous de nombreuses formes : elles peuvent être un ensemble d'intersections ou une série de vitesses que le robot doit prendre à chaque instant. Les informations dont le robot a besoin pour planifier sa trajectoire dépendent directement de l'environnement. Comprendre l'environnement, c'est connaître l'emplacement et la forme des obstacles [55].

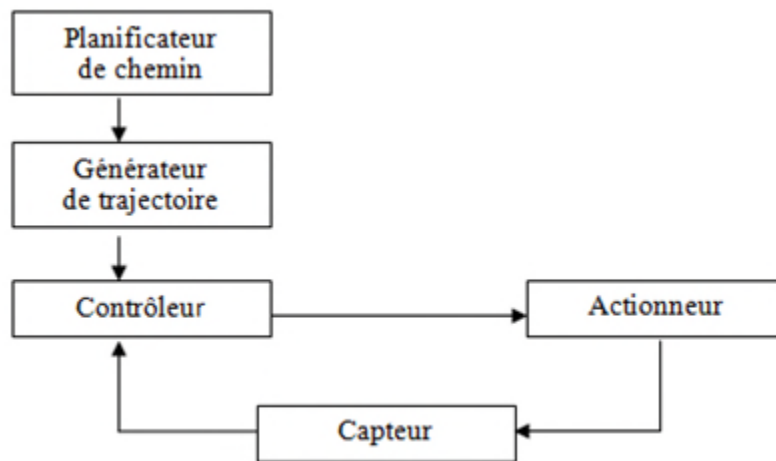


Figure III.3 : Planification et génération de trajectoires [56].

- Dans ce schéma, la première étape est relative à la planification d'un chemin traçant l'itinéraire désiré.

Dans la majorité des cas, elle consiste à prendre des décisions en fonction de la tâche à réaliser et de l'environnement de travail du robot. La deuxième étape est relative à la détermination et à la génération d'une trajectoire qui permet le déplacement du robot le long de son chemin. Les données de la planification sont exprimées sous la forme d'une trajectoire définie par une suite de repères constitués par les points de passage correspondant aux situations successives de l'effecteur. La dernière étape consiste à suivre cette trajectoire à partir de capteurs et d'actionneurs. Elle se résume souvent à une boucle d'asservissement qui permet la régulation de la trajectoire du robot par rapport à la trajectoire idéale [56].

III.3.4 Critères d'optimisation pour la planification des trajectoires

Différentes approches ont été proposées pour résoudre le problème de planification Des trajectoires. Ces approches sont conçues pour optimiser les critères formulés. Généralement, les critères à optimiser sont basés sur le temps d'exécution, l'énergie, le jerk ou la combinaison entre eux [53].

III.3.4.1 Minimisation du temps :

Dans l'industrie, les performances des systèmes robotiques tels que la productivité et la vitesse sont généralement prioritaires. Par conséquent, un critère important à prendre en compte lors de la planification des trajectoires est le temps d'exécution. En minimisant le temps d'exécution, la vitesse et la productivité peuvent être améliorées. Une option pour obtenir des trajectoires qui peuvent être exécutées en temps minimale consiste à approcher les chemins en utilisant des segments primitifs. De cette manière, le chemin entier est défini par les points intermédiaires reliant ces segments. Par conséquent, la façon la plus simple de trouver une trajectoire minimale est d'interpoler les points intermédiaires tout en satisfaisant certaines caractéristiques imposées comme la souplesse des entrées de commande du robot. Les approches les plus commun et largement utilisées pour cette effet sont basées sur l'optimisation des courbes splines où le temps d'exécution est la somme totale des intervalles de temps entre les points intermédiaires.

III.3.4.2 Minimisation de l'énergie :

La planification de trajectoire d'un robot en utilisant des critères énergétiques offre plusieurs avantages. D'une part, elle génère des trajectoires douces, plus facile à suivre, et permet de réduire la sollicitation des actionneurs et de la structure du manipulateur. En outre, les économies d'énergie peuvent être souhaitables dans certaines applications, telles que celles ayant une capacité limitée de la source d'énergie (par exemple des robots pour l'exploration spatiale ou sous-marine).

Les trajectoires point à point avec un minimum d'énergie sont en général considérées avec des bornes supérieures sur l'amplitude des signaux de commande et les vitesses articulaires [57].

III.3.4.3 Minimisation du jerk :

Une autre approche pour résoudre le problème de planification de chemin consiste à utiliser l'à-coup comme critère d'optimisation. Cette méthode est basée sur le fait qu'il existe une relation

Chapitre III : Recherche sur l'optimisation de la trajectoire d'un robot manipulateur à 2 DDI

entre le couple appliqué à l'actionneur et l'à-coup. Par conséquent, un moyen simple de satisfaire les contraintes de l'actionneur consiste à minimiser les à-coups. La minimisation des à-coups est très utile, notamment lorsqu'il s'agit de problèmes de suivi de trajectoire, car elle permet d'éviter les vibrations dans la structure mécanique. De plus, planification de trajectoire basée sur la minimisation de jerk. Produisez des paires régulières avec des ordres de grandeur acceptables. Cela limite les charges appliquées aux actionneurs et à la structure mécanique du robot [53].

III.3.5 Représentation mathématique de la trajectoire :

La représentation mathématique de la trajectoire du robot à 2DDL est essentielle pour la Planification et l'optimisation des mouvements. Une approche couramment utilisée consiste à décrire la position du robot en fonction du temps en utilisant des courbes paramétriques. Cette méthode permet de modéliser de manière précise et flexible les mouvements du robot dans l'espace.

Les courbes paramétriques décrivent la position du robot en fonction d'un paramètre, généralement le temps. Une courbe paramétrique bidimensionnelle pour un robot à 2DDL peut être définie comme suit :

$$x(t) = f_x(t)$$

$$y(t) = f_y(t)$$

où $x(t)$ et $y(t)$ représentent les coordonnées cartésiennes du robot à un instant t , $f_x(t)$ et $f_y(t)$ sont des fonctions paramétriques qui décrivent le déplacement du robot dans l'espace en fonction du temps.

Avantages des courbes paramétriques : Les courbes paramétriques offrent plusieurs avantages dans la représentation des trajectoires robotiques :

Flexibilité : Elles permettent de représenter une grande variété de trajectoires, des mouvements simples aux trajectoires complexes.

Facilité de manipulation : Les paramètres de la courbe peuvent être ajustés pour répondre à des objectifs spécifiques, tels que la vitesse, la courbure, etc.

Interpolation aisée : Elles facilitent l'interpolation entre points de passage, ce qui est essentiel pour des mouvements fluides.

III.4 Méthodes d'Optimisation de Trajectoire

III.4.1 Méthodes Classiques :

Dans cette sous-section, vous pouvez présenter différentes techniques d'optimisation utilisées pour trouver des trajectoires optimales pour les robots à 2DDL :

III.4.1.1 Méthode de Newton-Raphson :

La méthode de Newton-Raphson est une technique numérique largement utilisée pour résoudre des problèmes d'optimisation et de recherche de solutions approximatives à des équations non linéaires. Elle peut également être appliquée à la planification de trajectoire pour les robots à 2DDL, en particulier pour trouver des solutions aux équations de contraintes cinématiques et dynamiques. Voici une explication détaillée de la méthode de Newton-Raphson et de son application à la planification de trajectoire.

Principe : La méthode de Newton-Raphson est une méthode numérique qui permet de trouver les valeurs de la variable x pour laquelle $f(x) = 0$.

À partir d'une première estimation de la valeur de x , le développement en série de Taylor Autour d'une estimée x_0 nous donne [58]:

$$f(x) = f(x_0) + f'(x_0) \cdot (x - x_0) + f''(x_0) \cdot \frac{(x - x_0)^2}{2!}$$

Si x_0 est proche de la solution x de $f(x) = 0$, alors le carré de l'erreur $s_0 = (x - x_0)$ Et les termes de degré supérieur sont négligeables.

Sachant que :

$$f(x) = 0 \Rightarrow f(x_0) + f'(x_0) \cdot (x - x_0)$$

$$\varepsilon_0 = (x - x_0) = -\frac{f(x_0)}{f'(x_0)}$$

On peut donc considérer qu'une meilleure estimation de x sera :

$$x_1 = x_0 + \varepsilon_0$$

D'une manière générale :

$$x_{k+1} = x_k + \varepsilon_k$$

Chapitre III : Recherche sur l'optimisation de la trajectoire d'un robot manipulateur à 2 DDI

Avec:

$$\varepsilon_k = \frac{f(x_k)}{f'(x_k)}$$

k : est le numéro d itération

➤ Des équations précédentes, on obtient l'algorithme de Newton-Raphson.

$$x_{k+1} = x_k + \frac{f(x_k)}{f'(x_k)} \quad k=0, 1, 2$$

Il est intéressant de généraliser l'application de la méthode Newton-Raphson à la résolution d'un système de n équations non linéaires de la forme $F(x)=0$. Où:

$$f_1 = x_1, x_1, x_3 \cdots x_n = 0$$

$$f_2 = x_1, x_1, x_3 \cdots x_n = 0$$

⋮

$$f_m = x_1, x_1, x_3 \cdots x_n = 0$$

Et par analogie on calcul les valeurs des inconnues x a l itération (k+1) par la relation :

$$x_i^{(k+1)} = x_i^{(k)} + \Delta x_i^{(k)}$$

Le vecteur de correction $\Delta x_i^{(k)}$ est obtenu directement par le système d'équations linéaires :

$$j(x_i^{(k)}) \Delta x_i^{(k)} = -f(x_i^{(k)})$$

$j(x_i^{(k)})$: Est la matrice jacobéennes ou matrice des dérivées premières calculées à itération (k).

$$j(x_i^{(k)}) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$

Les éléments $\frac{\partial f_1}{\partial x_1}$ de la matrice Jacobienne sont calculés analytiquement en dérivant les fonctions $f(x_i^{(k)})$ par rapport $x_i^{(k)}$.

Le procédé de Newton-Raphson implique la résolution d'un système linéaire obtenu sous la

Forme : $A \cdot X = b$

Chapitre III : Recherche sur l'optimisation de la trajectoire d'un robot manipulateur à 2 DDI

A chaque itération, la solution peut être obtenue directement en inversant la matrice Jacobienne, c'est-à-dire :

$$\Delta x_i^{(k)} = [j(x_i^{(k)})]^{-1} \cdot F(x_i^{(k)})$$

Parce qu'il est difficile qu'à chaque itération de calculer inverse de la matrice au moment de Calcul des réseaux maillé on a remplacé seulement la variable x par le débit des tronçons.

Alors la relation devient :

$$Q_i^{(k+1)} = Q_i^{(k)} \frac{f(Q_i^{(k)})}{f'(Q_i^{(k)})}$$

➤ Application à la Planification de Trajectoire :

La méthode de Newton-Raphson peut être appliquée à la planification de trajectoire pour résoudre des problèmes de contraintes cinématiques et dynamiques. Voici comment elle peut être utilisée dans ce contexte :

Contraintes Cinématiques : Lors de la planification de trajectoire, les contraintes cinématiques, telles que les limitations de position et d'orientation des joints du robot, peuvent être exprimées sous forme d'équations non linéaires. La méthode de Newton-Raphson peut être utilisée pour résoudre ces équations et trouver des configurations articulaires valides pour le robot à chaque instant de la trajectoire.

Contraintes Dynamiques : Les contraintes dynamiques, telles que les forces et les moments maximaux que le robot peut supporter, peuvent également être exprimées sous forme d'équations non linéaires. La méthode de Newton-Raphson peut être utilisée pour résoudre ces équations et déterminer si la trajectoire planifiée respecte les contraintes dynamiques.

Optimisation de Trajectoire : En combinaison avec des techniques d'optimisation, la méthode de Newton-Raphson peut être utilisée pour ajuster la trajectoire planifiée afin de satisfaire les contraintes tout en optimisant des critères tels que le temps ou l'énergie.

Les Avantages et les Inconvénients de cette méthode :

Avantages :

- La méthode de Newton-Raphson est rapide et efficace pour converger vers des solutions locales.
- Elle est capable de gérer des contraintes non linéaires complexes.

Chapitre III : Recherche sur l'optimisation de la trajectoire d'un robot manipulateur à 2 DDI

Inconvénients :

- La méthode peut converger vers des solutions locales, ce qui signifie qu'elle peut ne pas trouver la solution globale si elle est loin de l'estimation initiale.
- Elle peut être sensible au choix de l'estimation initiale.

En résumé, la méthode de Newton-Raphson est une technique itérative puissante pour résoudre des équations non linéaires, y compris les contraintes cinématiques et dynamiques dans la planification de trajectoire pour les robots à 2DDL. Cependant, il est important de prendre en compte ses limitations, notamment en ce qui concerne la convergence vers des solutions locales.

III.4.1.2 La Méthode de descente de gradient :

La méthode de gradient est une approche itérative qui vise à trouver le minimum (ou le maximum) d'une fonction objectif en ajustant progressivement les paramètres de cette fonction. Elle repose sur le calcul du gradient de la fonction, qui indique la direction dans laquelle la fonction augmente le plus rapidement. Le processus d'itération est généralement défini par la formule suivante [59]:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha \nabla f(\mathbf{x}_k)$$

\mathbf{x}_k : est le vecteur de paramètres à l'itération n

α : est le taux d'apprentissage (ou pas d'apprentissage), qui détermine la taille des pas effectués à chaque itération.

$\nabla f(\mathbf{x}_k)$: est le gradient de la fonction objectif f évalué au point \mathbf{x}_k .

Le processus se répète jusqu'à ce qu'une condition d'arrêt soit satisfaite, par exemple lorsque la variation de la fonction objective devient négligeable.

Les points solution sont générés de proche en proche jusqu'à ce qu'un critère d'arrêt soit satisfait. Généralement, on utilise le critère suivant :

$$\|\nabla f(\mathbf{x}_k)\| \leq \varepsilon$$

ε : étant une valeur seuil qui indique la précision requise.

L'inconvénient de cette procédure est que la convergence est très dépendante du choix du pas de déplacement. La convergence peut être très lente si le pas est mal choisi. L'intérêt principal de cette méthode est de pouvoir se généraliser aux cas de fonctions non partout différentiables.

Chapitre III : Recherche sur l'optimisation de la trajectoire d'un robot manipulateur à 2 DDI

➤ **Application à l'Optimisation de Trajectoire :**

La méthode du gradient peut être utilisée pour optimiser des trajectoires de robots à 2DDL en ajustant les paramètres de la trajectoire de manière itérative pour atteindre un objectif spécifique, tel que la minimisation du temps ou de l'énergie. Voici comment elle peut être appliquée dans ce contexte :

Fonction Objectif : Définissez une fonction objective qui mesure la performance de la trajectoire en fonction du critère spécifique que vous souhaitez optimiser. Par exemple, pour la minimisation du temps, la fonction objective pourrait être le temps total de parcours de la trajectoire.

Paramètres de la Trajectoire : Identifiez les paramètres de la trajectoire qui peuvent être ajustés pour optimiser la fonction objective. Cela pourrait inclure des paramètres tels que les positions et les vitesses initiales et finales, les points de passage, ou d'autres caractéristiques de la trajectoire.

Descente de Gradient : Utilisez la méthode du gradient pour ajuster progressivement les paramètres de la trajectoire afin de minimiser ou maximiser la fonction objective. À chaque itération, calculez le gradient de la fonction objectif par rapport aux paramètres de la trajectoire et ajustez ces paramètres en conséquence.

Condition d'Arrêt : Définissez une condition d'arrêt appropriée pour le processus d'optimisation, par exemple en arrêtant les itérations lorsque la variation de la fonction objective devient négligeable ou lorsque le nombre d'itérations maximum est atteint.

- Le choix du taux d'apprentissage (α) peut affecter la convergence et la stabilité de l'algorithme.

III.4.2 Algorithmes d'Optimisation Heuristiques :

Les algorithmes d'optimisation heuristiques sont des méthodes de résolution de problèmes qui visent à trouver des solutions de haute qualité pour des problèmes complexes en utilisant des approches heuristiques, c'est-à-dire des règles empiriques ou des techniques inspirées de l'intelligence humaine pour guider la recherche. Ces algorithmes sont particulièrement utiles pour résoudre des problèmes d'optimisation difficiles où la recherche exhaustive de toutes les solutions possibles n'est pas réalisable en raison de contraintes de temps ou de ressources. Voici quelques exemples d'algorithmes d'optimisation heuristiques et comment ils peuvent être utilisés dans le contexte de la planification de trajectoire pour les robots à 2DDL :

Chapitre III : Recherche sur l'optimisation de la trajectoire d'un robot manipulateur à 2 DDI

III.4.2.1 Les algorithmes génétiques :

- Principe d'un algorithme génétique :

En 1975, John Holland et ses collègues ont développé les algorithmes génétiques, s'inspirant du processus d'évolution biologique pour créer des méthodes d'optimisation artificielle. Ces algorithmes imitent la sélection naturelle, la réplication et d'autres mécanismes biologiques. Ils utilisent des populations de solutions potentielles pour résoudre des problèmes. Chaque solution est représentée par un individu avec un ou plusieurs chromosomes, contenant des gènes. Les informations génétiques sont modifiées par le croisement ou la mutation, et les individus les plus adaptés sont sélectionnés pour former la génération suivante. Ce processus se répète jusqu'à ce que des solutions satisfaisantes soient trouvées. Les algorithmes génétiques artificiels reflètent ainsi les principes de l'évolution naturelle pour résoudre des problèmes complexes [53].

- Mécanismes d'un algorithme génétique :

Les algorithmes génétiques sont basés sur des processus d'évolution similaire des mécanismes de la génétique naturelle. Ces mécanismes concernent la reproduction génétique, le croisement et la mutation.

- Reproduction génétique :

Le processus de reproduction dans les algorithmes génétiques est une adaptation artificielle du concept biologique de reproduction darwinienne. Dans la reproduction naturelle, les organismes luttent pour survivre et se reproduire, résistant aux obstacles tels que les maladies et les prédateurs. Certains individus prospèrent tandis que d'autres souffrent. Dans les algorithmes génétiques, cette lutte pour la survie est reflétée par la mesure de la valeur de la fonction objectif des individus. Les individus avec une meilleure valeur de fonction objectif ont plus de chances de survivre, tandis que ceux avec une valeur de fonction objectif plus faible sont éliminés. Les individus favorisés sont alors reproduits, et leurs informations génétiques sont transmises aux générations suivantes, simulant ainsi le processus d'évolution naturelle pour résoudre des problèmes complexes.

- Croisement :

L'opérateur de reproduction permet de sélectionner des individus favorisés et de reproduire exactement leur information génétique. L'opération suivante, le croisement, combine l'information génétique des individus sélectionnés pour créer les enfants de la nouvelle

Chapitre III : Recherche sur l'optimisation de la trajectoire d'un robot manipulateur à 2 DDI

Génération. Cette opération consiste à échanger des parties de matériel génétique entre individus parents pour permettre la création d'enfants. C'est une autre imitation de l'opération naturelle. Les enfants artificiels, comme les biologiques, combinent les qualités de leur parent [53].

- Mutation :

La mutation représente une modification aléatoire de portions de l'information génétique des individus privilégiés. Son rôle est essentiel pour garantir une exploration globale lors du processus d'optimisation. Au cours des phases de reproduction et de croisement, des individus sont sélectionnés et leurs informations génétiques peuvent parfois se ressembler. Dans de tels cas, l'algorithme de recherche peut ne pas garantir la convergence vers un optimum de meilleure qualité. Afin de remédier à ce problème potentiel, la mutation est employée en tant qu'opération complémentaire visant à prévenir l'homogénéité au sein de la population. Par conséquent, lorsque l'algorithme génétique maintient des altérations aléatoires, la recherche tend à aboutir à des résultats optimaux. En somme, la mutation introduit de la variabilité au sein de la population, ce qui se révèle essentiel pour l'efficacité des algorithmes génétiques lors de la quête de solutions optimales [53].

➤ L'optimisation des trajectoires par les algorithmes génétiques consiste à utiliser ces algorithmes pour trouver des trajectoires optimales dans des problèmes de planification de mouvement. Voici comment cela fonctionne en général :

- Représentation des Trajectoires : Tout d'abord, les trajectoires du robot sont représentées sous forme de gènes, où chaque gène code un aspect de la trajectoire, par exemple, les positions et les vitesses des articulations du robot à différents instants.
- Population Initiale : Une population initiale de trajectoires est générée de manière aléatoire. Chaque trajectoire est un individu de cette population.
- Fonction Objectif : Une fonction objective est définie pour évaluer la qualité des trajectoires en fonction des critères spécifiques, tels que la minimisation du temps, de l'énergie, ou d'autres objectifs pertinents.
- Évolution : Les algorithmes génétiques utilisent des opérations telles que la sélection, le croisement (recombinaison) et la mutation pour faire évoluer la population de trajectoires au fil des générations. Les trajectoires de meilleure qualité ont une probabilité plus élevée d'être sélectionnées, tandis que le croisement et la mutation introduisent de la diversité dans la population.

Chapitre III : Recherche sur l'optimisation de la trajectoire d'un robot manipulateur à 2 DDI

- Convergence : Au fur et à mesure que les générations passent, les trajectoires tendent à s'améliorer en fonction de la fonction objective, convergeant vers une solution optimale ou près de l'optimum.
- Résultats Optimaux : Une fois que l'algorithme atteint un critère d'arrêt (par exemple, un certain nombre de générations ou une amélioration minimale), il retourne la meilleure trajectoire trouvée comme résultat.



Figure. III.4 : Représentation de l'opération de mutation

- **La Programmation Génétique (Genetic Programming : GP)** consiste à faire évoluer des Structures d'arbres représentant des programmes [60].

Algorithme

1. initialisation :

- Initialiser le compteur des générations $t=0$.
- Initialiser aléatoirement une population $P(t)$ de taille fixée N .

Evaluer chaque individu de $P(t)$.

2. Sélectionner les individus les plus performants (au sens de F) de la population, les recopier pour former une nouvelle population de même taille.

3. Créer une nouvelle population en appliquant les opérateurs de variation :

- Opérateur de croisement : recombinaison des parties de deux individus pour en obtenir deux nouveaux.
- Opérateur de mutation : modifier aléatoirement un individu.

4. Evaluer le fitness de chaque individu de la nouvelle population.

5. Remplacer certains individus de l'ancienne population par les meilleurs individus de la nouvelle population.

6. Si la condition d'arrêt n'est pas vérifiée, aller vers l'étape 2, sinon, retourner le meilleur individu de $P(t)$.

Chapitre III : Recherche sur l'optimisation de la trajectoire d'un robot manipulateur à 2 DDI

III.4.2.2 Optimisation par Essaim Particulaire (PSO) :

Les chercheurs se sont inspirés du comportement des essaims d'oiseaux et des espèces de poissons pour développer des méthodes d'optimisation, notamment « Kennedy et Eberhart ». Initialement, leur intérêt était de reproduire la manière dont les oiseaux se déplacent vers leur perchoir connu, mais ils ont été confrontés au défi de comprendre comment les oiseaux découvrent des sources de nourriture imprévisibles. Cela a marqué un tournant, car cela les a amenés à passer de l'imitation du comportement à la résolution de problèmes d'optimisation. Inspirés par le comportement de recherche de nourriture des oiseaux et des poissons, « Kennedy et Shi » ont proposé l'approche de l'optimisation par les essaims particulaires. C'est un paradigme basé sur la recherche de nourriture artificielle, où les agents modélisés suivent le même principe que les oiseaux et les poissons dans leur recherche naturelle de nourriture. L'algorithme génère initialement un ensemble de solutions potentielles appelées "particules" et évalue leur qualité à l'aide d'une fonction de coût. À l'instar de leurs homologues naturels, les particules artificielles ajustent leur position et leur vitesse pour suivre les autres membres de la population. Ce processus se répète jusqu'à ce que la solution optimale soit trouvée. Une fois cette solution trouvée, elle attire d'autres particules, fournissant ainsi non seulement une solution optimale mais également un ensemble de solutions.

Cette approche s'inspire du comportement collectif d'exploration et d'optimisation observé dans les essaims naturels pour résoudre des problèmes d'optimisation complexes. Elle est utilisée dans divers domaines, notamment en intelligence artificielle et en recherche opérationnelle, pour résoudre une gamme de problèmes d'optimisation [61].

➤ Optimisation de trajectoire par PSO:

On commence par créer un ensemble de trajectoires faisables (c.à.d. trajectoires qui relient le point de départ et le point d'arrivée et qui évitent la collision avec les obstacles), chaque trajectoire est créée à partir de la trajectoire de base (trajectoire obtenue par ACO ou A*) en modifiant aléatoirement les coordonnées de chaque point de cette trajectoire tout en restant au voisinage de ce point pour éviter la collision avec les obstacles [55].

$$\begin{cases} x_b^i + \varepsilon \leq x_g^i & \leq x_g^i + \varepsilon \\ x_b^i - \varepsilon \leq x_g^i & \leq x_g^i + \varepsilon \end{cases}$$

x_b^i L'abscisse du point i de la trajectoire de base.

x_g^i L'abscisse du point i de la trajectoire générée.

Chapitre III : Recherche sur l'optimisation de la trajectoire d'un robot manipulateur à 2 DDI

y_b^i L'ordonnée du point i de la trajectoire de base.

y_b^i L'ordonnée du point i de la trajectoire généré.

ε Un réel positive qui permet de contrôler le voisinage de changement des coordonnées (Il doit être petit pour éviter la collision avec les obstacles).

Après avoir obtenu l'ensemble de trajectoires, on applique les principes de l'algorithme PSO pour trouver une trajectoire optimale (chaque particule représente une trajectoire Composé par un ensemble de point). Le critère d'optimisation ou la fonction objectif est la longueur du chemin.

- Initialisation de l'algorithme:

On initialise les particules par les trajectoires créés à partir de la trajectoire base créée par ACO ou A*, et on initialise aussi P_{best}^i , P_{best}^g , la vitesse de chaque particule et comme ce dernier est une trajectoire (vecteur de dimension n ————— 2) donc la vitesse est aussi un vecteur de même dimension.

- P_{best}^i best meilleure trajectoire trouvée par la particule i.
- P_{best}^g best meilleure trajectoire trouvée par toutes les particules.

- Pseudo code de l'algorithme :

```
Initialisation des positions et des vitesses de chaque particule.
Pour itération = 1, 2...N
  Pour particule = 1, 2...P
    Déplacement de la particule suivant l'équation du mouvement.
    Pour k = 1 : dimension de la particule
      Si (le point k cas sort de l'espace de recherche) alors
        Remplacer le point k par le point le plus proche situé sur le limite.
    Fin de si
  Fin de pour
  Fin de pour
  Mise à jour de :  $P_{best}^i$ ,  $P_{best}^g$ 
Fin de pour
Remplacez la trajectoire de base par  $P_{best}^g$  trouvée.
```

Chapitre III : Recherche sur l'optimisation de la trajectoire d'un robot manipulateur à 2 DDI

III.4.2.3 Recuit simulé :

En physique de la matière condensée, deux processus différents peuvent être distingués en analysant l'effet de la régulation de la température sur l'état final du matériau. D'une part, la trempe est le processus consistant à chauffer un matériau en élevant sa température jusqu'à un niveau où il devient liquide, puis à le refroidir rapidement pour former un solide. Dans un premier temps, les atomes du liquide sont répartis de manière aléatoire en raison de la température élevée. Un refroidissement rapide n'offre aucune stabilité thermique. L'état final du matériau est donc un solide amorphe métastable. Le recuit, quant à lui, est un processus consistant à abaisser progressivement la température plutôt qu'à un refroidissement rapide. Semblable à la trempe, le matériau fondu à haute température est constitué d'un grand nombre d'atomes non structurés. Cependant, en refroidissant lentement le liquide et en permettant au matériau d'atteindre l'équilibre thermique, l'état final est un cristal solide avec une énergie minimale.

Le plus ancien algorithme de simulation de l'équilibre thermique d'un ensemble de particules a été proposé par Metropolis. Initialement, les particules sont placées dans un réseau régulier à une température donnée. Toutes les particules dans leur configuration originale se déplacent alors continuellement dans le plan. Au cours de chaque cycle, une particule donnée est soumise à un petit déplacement arbitraire et la variation de l'énergie du système est calculée. A ce stade, lorsque l'énergie du système diminue, les nouvelles positions des particules sont acceptées. Cependant, lorsque l'énergie du système augmente, de nouvelles positions des particules peuvent également être acceptées. La probabilité d'accepter ce mouvement est égale à $\exp(-\Delta E/K_B T)$. Par conséquent, lorsque la probabilité d'acceptation est supérieure à la probabilité générée aléatoirement, la nouvelle position est autorisée ; sinon, l'ancienne est conservée. Enfin, le système continue d'évoluer, suivant la distribution de Boltzmann, jusqu'à ce que l'équilibre thermique soit atteint.

Le recuit simulé des problèmes d'optimisation fonctionne de manière similaire à l'algorithme Metropolis, qui imite le processus d'un système physique.

L'énergie du système physique est remplacée par la fonction objectif, tandis que la configuration est remplacée par les paramètres de la fonction objectif. Par conséquent, simuler le processus de recuit pour un problème donné nécessite d'augmenter la température jusqu'à ce qu'elle atteigne le niveau de fusion et de l'abaisser progressivement jusqu'à atteindre le niveau de

Chapitre III : Recherche sur l'optimisation de la trajectoire d'un robot manipulateur à 2 DDI

solidification. A chaque température, la configuration d'origine est perturbée et la valeur de la fonction objectif est calculée. Lorsqu'elle diminue, la nouvelle configuration est acceptée. Cependant, à mesure que la valeur de la fonction objective augmente, si la probabilité est supérieure à toute probabilité, la nouvelle configuration est acceptée, sinon elle est rejetée. Ce cycle se poursuit jusqu'à ce que l'équilibre thermique soit atteint. La configuration du système en équilibre thermique est considérée comme une solution au problème d'optimisation en suivant ce processus jusqu'à des températures glaciales.

- Planification des trajectoires par recuit simulé :

L'algorithme de recuit simulé est appliqué à des problèmes de planification de trajectoire pour différents types de robots et dans diverses situations. Voici des principales conclusions de chaque étude :

- 1.Garg** utilise deux techniques d'optimisation différentes pour résoudre des problèmes de planification de trajectoire. Dans le premier cas, il cherche à trouver un chemin optimal avec un couple minimal pour un manipulateur à deux degrés de liberté. Dans le deuxième cas, il étudie la planification de trajectoires pour deux robots coopérants, chacun ayant deux degrés de liberté, tout en tenant compte des forces exercées par l'un sur l'autre. Les fonctions objectives prennent en compte l'ensemble des paramètres pertinents, et Garg utilise à la fois des algorithmes génétiques et le recuit simulé adaptatif pour résoudre ces problèmes. Les résultats montrent que le recuit simulé adaptatif est plus performant en termes de convergence par rapport aux algorithmes génétiques.
- 2.: Peng** propose une approche hybride combinant l'algorithme génétique et le recuit simulé pour déterminer une trajectoire optimale pour un manipulateur planaire à trois degrés de liberté. Les opérateurs génétiques sont adaptés pour prendre en compte des critères spécifiques, tels que la génération de l'individu, les valeurs extrêmes de la fonction objectif et le nombre maximum de générations. Le recuit simulé est utilisé pour choisir entre les individus parents et enfants en fonction de la valeur de la fonction objectif. Cette approche conduit à une minimisation du déplacement total des articulations du robot, et les performances de l'algorithme hybride surpassent celles de l'algorithme génétique standard.
- 3.Miao** applique l'algorithme de recuit simulé à la planification de chemin d'un robot mobile dans un environnement dynamique. Deux paradigmes de planification des trajectoires sont utilisés : une planification hors ligne basée sur des informations statiques sur les obstacles et une planification en ligne qui ajuste la trajectoire en fonction des obstacles mobiles détectés par des

Chapitre III : Recherche sur l'optimisation de la trajectoire d'un robot manipulateur à 2 DDI

Capteurs embarqués. L'algorithme de recuit simulé est utilisé pour générer des trajectoires optimales en prenant en compte les obstacles et les contraintes de l'environnement. Les performances de cette technique sont évaluées dans différents environnements.

4.Jianjun utilise l'algorithme de recuit simulé pour trouver une trajectoire à temps optimal pour un robot à roues à quatre degrés de liberté. Le robot est modélisé de manière simplifiée, et les contraintes sur les variables articulaires sont gérées en tant que fonctions de pénalité dans la fonction objective. L'algorithme de recuit simulé est appliqué pour minimiser le temps de parcours en générant des trajectoires optimales. Le processus itératif de l'algorithme réduit progressivement la température jusqu'à atteindre une température finale. La trajectoire optimale est trouvée avec un temps d'exécution inférieur à trois secondes.

III.4.2.4 Recherche tabou :

La recherche tabou est une approche métaheuristique proposée par Glover [Glover. Ce mécanisme enregistre les déplacements vers les solutions récemment examinées et les marque comme mouvements interdits ou tabous. Généralement, la restriction des mouvements vers des solutions antérieures est flexible dans le sens où ces solutions ont un statut tabou temporaire et peut donc être visité à nouveau. De cette façon, la recherche tabou commence par l'utilisation du même principe que les méthodes de gradient, mais, dès que la recherche est coincée dans un optimum local, il favorise le choix d'aller vers un mauvais solution en visitant une solution précédemment examinée [53].

III.4.2.4.1 Principes de la recherche tabou :

Un moyen efficace de faciliter la mise en œuvre de la recherche tabou, après le succès obtenu, consiste à décrire les principes fondamentaux de l'approche. Ces principes sont établis par Glover où les plus importants sont : la mémoire à court terme, la mémoire à long terme et le critère d'aspiration.

- Mémoire à court terme : Pendant le processus de recherche, les itinéraires vers les solutions visitées sont stockés dans une liste appelée liste tabou. De manière générale, la longueur de cette liste (appelée onglet « tenure ») détermine le type de mémoire utilisé. La mémoire à court terme signifie qu'un nombre limité d'actions taboues peuvent être stockées. Du coup, les mémoires sont vite occupées et le statut tabou de ces voyages s'oublie facilement. La mémoire à court terme est souvent utilisée pour diriger les processus de recherche vers des zones locales. Cette stratégie est appelée intensification.

Chapitre III : Recherche sur l'optimisation de la trajectoire d'un robot manipulateur à 2 DDI

- Mémoire à long terme : Contrairement à la mémoire à court terme, qui vise à intensifier la recherche des régions locales, la mémoire à long terme vise à diversifier la recherche vers des régions non découvertes. Une façon de favoriser la diversification est de conserver le statut tabou des déplacements pénalisés pour une période beaucoup plus longue. L'augmentation de la longueur de la mémoire aide à guider le processus de recherche vers les régions non explorées.

- Critère d'aspiration : Puisque la recherche tabou est basé sur la définition des mouvements récents comme des mouvements interdits, certains de ces mouvements sont autorisés lorsqu'ils améliorent la valeur de la fonction objective. Cette autorisation est connue sous le nom du critère d'aspiration.

III.4.2.4.2. Planification de trajectoire à l'aide de la recherche taboue :

L'algorithme de recherche tabou ainsi que d'autres métaheuristiques pour résoudre des problèmes de planification de trajectoires pour des robots mobiles dans des environnements divers. Voici un résumé des principales conclusions de chaque étude [53] :

Masehian: Masehian utilise l'approche de recherche tabou pour concevoir un planificateur de mouvement en ligne pour un robot mobile. L'algorithme de recherche tabou est basé sur la perception du robot de son environnement à travers des capteurs. Il utilise une mémoire à court terme et une mémoire à long terme pour éviter de revisiter des zones déjà explorées. L'algorithme favorise les déplacements à faible coût le long des sommets d'obstacles pour générer une trajectoire optimale. Lorsque le robot est piégé dans une impasse, la diversification est activée. Les résultats montrent l'efficacité de l'algorithme dans le traitement de différentes situations.

Hussein: Hussein mène une étude comparative entre les métaheuristiques basées sur des individus (recuit simulé, recherche tabou) et les métaheuristiques basées sur des populations (algorithme génétique) pour résoudre le problème de planification de trajectoires dans un environnement de grille. Les algorithmes sont évalués en termes de temps nécessaire pour trouver le chemin optimal, le nombre d'itérations requis, le temps par itération et la qualité du chemin trouvé. La recherche tabou est efficace en termes de nombre d'itérations, tandis que le recuit simulé est plus rapide que l'algorithme génétique pour trouver le chemin optimal.

Imen: Imen étudie l'efficacité de l'approche de recherche tabou pour la planification globale du chemin d'un robot mobile dans un environnement de grille. La recherche tabou est appliquée aux sous-chemins composant le chemin initial pour réduire le temps d'exploration. Les listes taboues sont utilisées pour éviter de revenir aux mêmes déplacements. Un mécanisme

Chapitre III : Recherche sur l'optimisation de la trajectoire d'un robot manipulateur à 2 DDI

d'aspiration est parfois activé pour améliorer la fonction objective. L'algorithme de recherche tabou est testé et comparé à d'autres algorithmes dans différents environnements de grille.

Panda : Panda propose une approche hybride combinant l'algorithme des essais particuliers et la recherche tabou pour résoudre le problème de la planification de trajectoires. Initialement, l'algorithme des essais particuliers est utilisé pour explorer des chemins potentiels. Ensuite, la recherche tabou est employée pour améliorer ces chemins à la fin de chaque génération. Cette approche hybride est comparée à d'autres algorithmes pour la planification de trajectoires avec plusieurs robots dans un environnement statique. Les résultats montrent que l'approche hybride est plus performante en termes de convergence et de qualité des solutions.

En résumé, ces études montrent que l'algorithme de recherche tabou est une méthode efficace pour résoudre des problèmes de planification de trajectoires pour des robots mobiles. Il peut être utilisé seul ou combiné avec d'autres métaheuristiques pour obtenir des solutions de haute qualité dans divers environnements et situations.

III.5 Conclusion :

Le chapitre III a exploré en profondeur les aspects de l'optimisation de trajectoire pour les robots manipulateurs à 2DDL (2 degrés de liberté). Nous avons examiné divers éléments clés liés à ce domaine, notamment la définition formelle du problème, les types de trajectoires à optimiser, et les méthodes d'optimisation couramment utilisées. Nous avons commencé par les méthodes classiques (Méthode de descente de gradient et Newton-Raphson) ensuite les métaheuristiques à base population qui sont les algorithmes génétiques, les essaims particuliers. Ensuite, nous avons abordé les métaheuristiques à base individu qui sont le recuit simulé et la recherche tabou. Dans le chapitre suivant, nous nous consacrerons à la programmation, à la simulation et à l'interprétation des résultats.

Chapitre IV

Recherche sur les algorithmes PSO (particule swarm optimization), simulation et interprétation du résultat

IV.1 Introduction

Dans ce chapitre nous allons développer l'optimisation des trajectoires d'un robot manipulateur planaire à 2ddl, en appliquant l'algorithme essaim des particules.

Le travail consiste à diviser la trajectoire en deux sous trajectoires, la première relie le point initial au point intermédiaire, elle est représentée par une spline quadratique (un polynôme de degré 4) et la seconde relie ce point au point final, elle est représentée par une spline quintique (un polynôme de degré 5).

Sur chaque trajectoire est optimisé les profils de position, vitesse et accélération ainsi qu'une minimisation d'énergie en limitant les couples des actionneurs.

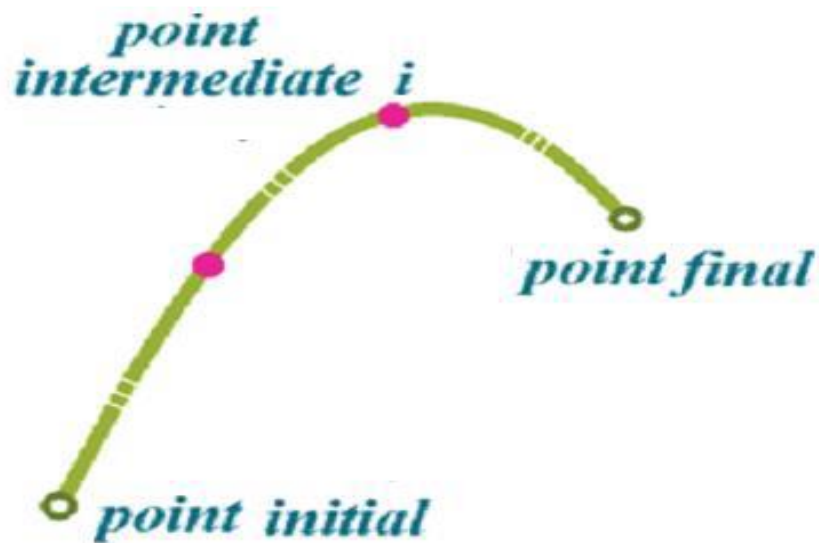


Figure IV.1 : Trajectoire point par point.

Entre le point initial et le point intermédiaire on utilise le spline quadratique suivant :

$$\theta_{i,i+1}(t) = a_{i0} + a_{i1}T_i + a_{i2}T_i^2 + a_{i3}T_i^3 + a_{i4}T_i^4, (i = 0, \dots, mp - 1) \quad (\text{IV. 1})$$

Où (a_{i0}, \dots, a_{i4}) sont des constantes qui sont déterminées par les contraintes suivantes :

$$\theta_i = a_{i0} \quad (\text{IV. 2})$$

$$\theta_{i+1} = a_{i0} + a_{i1}T_i + a_{i2}T_i^3 + a_{i4}T_i^4 \quad (\text{IV. 3})$$

$$\dot{\theta}_i = a_{i1} \quad (\text{IV. 4})$$

$$\dot{\theta}_{i+1} = a_{i1} + 2a_{i2}T_i + 3a_{i3}T_i^2 + 4a_{i4}T_i^3 \quad (\text{IV. 5})$$

Chapitre IV : Recherche sur les algorithmes PSO (particule swarm optimization), simulation et interprétation du résultat

$$\ddot{\theta}_i = 2a_{i2} \quad (IV. 6)$$

Où T_i représente le temps d'exécution du point i au point $i+1$.

- La solution des équations précédentes nous donne les cinq inconnus par :

$$a_{i0} = \theta_i \quad (IV. 7)$$

$$a_{i1} = \dot{\theta}_i \quad (IV. 8)$$

$$a_{i2} = \ddot{\theta}_i / 2 \quad (IV. 9)$$

$$a_{i3} = (4\theta_{i+1} - \dot{\theta}_{i+1}T_i - 4\theta_i - 3\ddot{\theta}_iT_i^2) / T_i^3 \quad (IV. 10)$$

$$a_{i4} = (\dot{\theta}_{i+1}T_i - 3\theta_{i+1} + 3\theta_i + 2\dot{\theta}_iT_i + \ddot{\theta}_iT_i^2 / 2) / T_i^4 \quad (IV. 11)$$

- L'accélération intermédiaire du point $i+1$ est donnée par :

$$\ddot{\theta}_{i+1} = 2a_{i2} + 6a_{i3}T_i + 12a_{i4}T_i^2 \quad (IV. 12)$$

- La trajectoire qui relie le point intermédiaire au point final est représentée par le spline quintique suivant :

$$\theta_{i,i+1}(t) = b_{i0} + b_{i1}T_i + b_{i2}T_i^2 + b_{i3}T_i^3 + b_{i4}T_i^4 + b_{i5}T_i^5, (i = mp) \quad (IV. 13)$$

Où $(b_{i0} \dots b_{i5})$ sont des constantes qui sont déterminées par les contraintes suivantes :

$$\theta_i = b_{i0} \quad (IV. 14)$$

$$\theta_{i+1} = b_{i0} + b_{i1}T_i + b_{i2}T_i^2 + b_{i3}T_i^3 + b_{i4}T_i^4 + b_{i5}T_i^5 \quad (IV. 15)$$

$$\dot{\theta}_i = b_{i1} \quad (IV. 16)$$

$$\dot{\theta}_{i+1} = b_{i1} + 2b_{i2}T_i + 3b_{i3}T_i^2 + 4b_{i4}T_i^3 + 5b_{i5}T_i^4 \quad (IV. 17)$$

$$\ddot{\theta}_i = 2b_{i2} \quad (IV. 18)$$

$$\ddot{\theta}_{i+1} = 2b_{i2} + 6b_{i3}T_i + 12b_{i4}T_i^2 + 20b_{i5}T_i^3 \quad (IV. 19)$$

En addition, la solution des équations précédentes nous donne les cinq inconnus par :

$$b_{i0} = \theta_i \quad (IV. 20)$$

$$b_{i1} = \dot{\theta}_i \quad (IV. 21)$$

$$b_{i2} = \ddot{\theta}_i / 2 \quad (IV. 22)$$

$$b_{i3} = (20\theta_{i+1} - 20\theta_i - (8\dot{\theta}_{i+1} + 12\dot{\theta}_i)T_i - (3\ddot{\theta}_i - \ddot{\theta}_{i+1})T_i^2) / 2T_i^3 \quad (\text{IV. 23})$$

$$b_{i4} = (30\theta_{i+1} - 30\theta_i - (14\dot{\theta}_{i+1} + 16\dot{\theta}_i)T_i - (3\ddot{\theta}_i - 2\ddot{\theta}_{i+1})T_i^2) / 2T_i^4 \quad (\text{IV. 24})$$

$$b_{i5} = (12\theta_{i+1} - 12\theta_i - (6\dot{\theta}_{i+1} + 6\dot{\theta}_i)T_i - (\ddot{\theta}_i - \ddot{\theta}_{i+1})T_i^2) / 2T_i^5 \quad (\text{IV. 25})$$

Au total les paramètres à déterminer sont les positions et les vitesses des articulations aux points intermédiaires et les temps d'exécutions de chaque sous trajectoire, soit un nombre de six paramètres.

IV.2 Planification de trajectoires par l'algorithme essaim des particules :

L'algorithme du PSO doit nous fournir une trajectoire optimale ayant un espace minimal et un temps minimal tout à ne pas dépasser le couple maximal prédéfini pour les actionneurs. La planification des trajectoires adopte le modèle géométrique direct du robot pour éviter les singularités.

Pour le robot planaire 2R à 2ddl il y en a six paramètres à optimiser :

$$\left[q_1 \quad q_2 \quad \dot{q}_1 \quad \dot{q}_2 \quad t_1 \quad t_2 \right] \quad (\text{IV. 26})$$

Où q_i et \dot{q}_i sont les positions et les vitesses angulaires des deux articulations du robot.

t_1 : est le temps d'exécutions du point initial au point intermédiaire.

t_2 : est le temps d'exécutions du point intermédiaire au point final.

IV.3 Paramètres de l'algorithme PSO :

Le nombre d'individus (oiseaux) du groupe est de 20, leurs positions et vitesse initiales sont générées aléatoirement, les vitesses cognitives et les poids d'adaptation sont choisis par défaut par le nombre 2, le poids d'inertie minimal est choisi de 0.4 et le poids d'inertie maximal est choisi de 0.9, ce sont des valeurs souvent choisis. Le nombre d'itération est fixé à 50 et le critère de la fonction objective est décrit dans le paragraphe suivant.

IV.4 Critère de la fonction objective :

Quatre indices sont utilisés pour qualifier l'évolution de la trajectoire libre dans l'espace de travail pour les robots manipulateurs. Tous les indices sont traduits en fonctions de la pénalité à minimiser. Chaque indice est calculé individuellement et est intégré dans l'évaluation de la fonction objective physique. Cette fonction notée f_f qui est adoptée pour évaluer les trajectoires candidates se définit comme suit :

$$f_f = \beta_1 f_{ot} + \beta_2 f_q + \beta_3 f_c + \beta_4 t_T \quad (IV. 27)$$

▪ L'objectif d'optimisation consiste à trouver un ensemble de paramètres de conception qui minimisent la fonction objective f_f selon les priorités données par les facteurs de pondération $\beta_i(i, \dots, 4)$ où chaque ensemble différent de facteurs de pondération donne une solution différente.

L'indice f_{ot} représente la quantité excessive conduite, par rapport au couple maximal τ_{max} , qui est demandé par le $i^{ème}$ moteur d'articulation sous la trajectoire considérée donnée par :

$$f_{ot} = \sum_{j=1}^b \sum_{i=1}^a f_i^j$$

$$f_i^j = \begin{cases} 0 & |\tau_i^j| < \tau_{i\ max} \\ |\tau_i^j| - \tau_{i\ max} & otherwise \end{cases} \quad (IV. 28)$$

Où a est le nombre d'articulations du robot et b est le nombre de sous trajectoires de position d'un point à un autre.

Le modèle dynamique du robot planaire 2R est établi par les équations de Newton-Euler.

Le Critère f_q représente la distance totale parcourue par les 2 articulations du robot, il est donné par :

$$f_q = \sum_{j=1}^b \sum_{i=1}^a |q_{ij} - q_{ij-1}| \quad (IV. 29)$$

Le critère f_c représente la longueur cartésienne totale parcourue par le robot, il est donné par :

$$f_c = \sum_{j=2}^b d(p_j, p_{j-1}) \quad (IV.30)$$

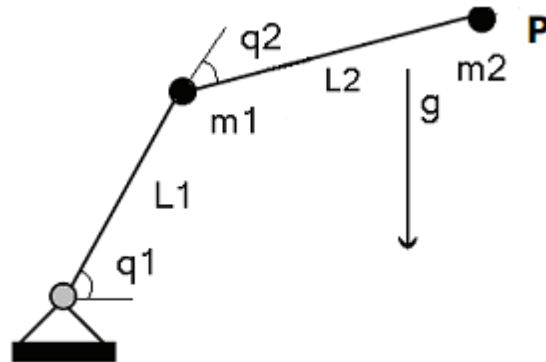
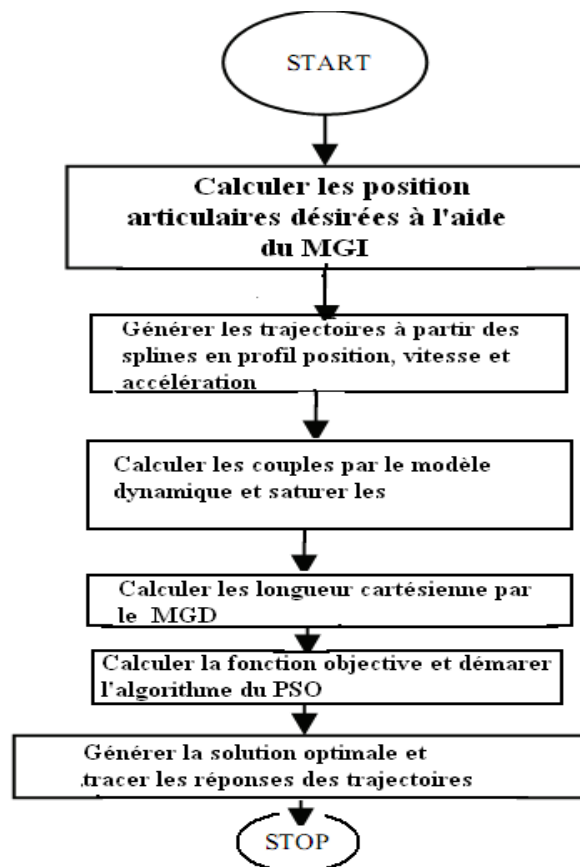


Figure IV.2 : Robot planaire 2R 2ddl.

➤ Diagramme d'optimisation des trajectoires par PSO :



IV.5 Résultats des simulations :

1^{er} Essai :

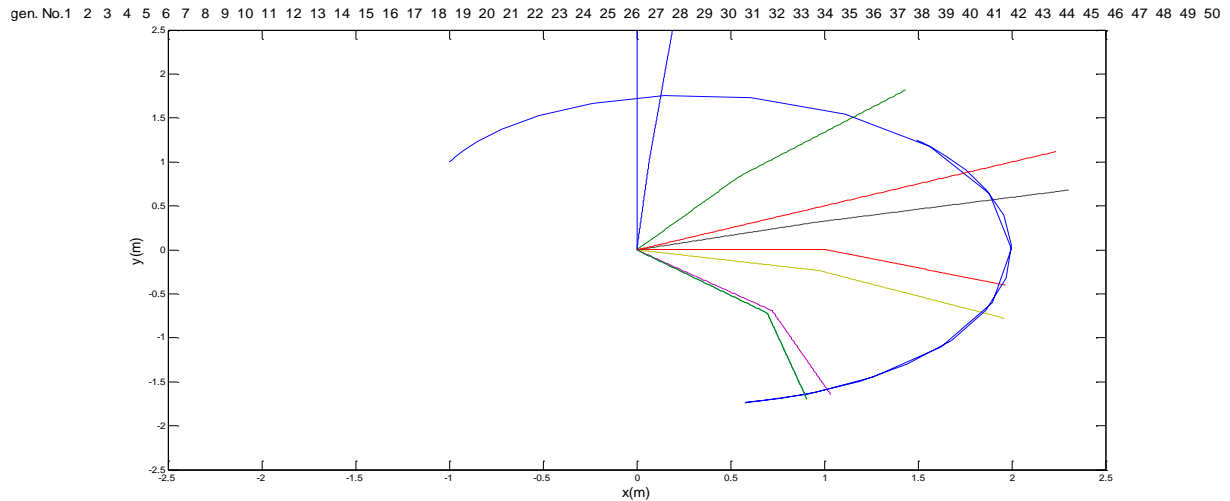


Figure IV.3 : Trajectoire du robot dans l'espace cartésien.

○ La figure IV.3 montre la trajectoire cartésienne optimale du robot planaire reliant les points initial et final dans l'espace opérationnel,

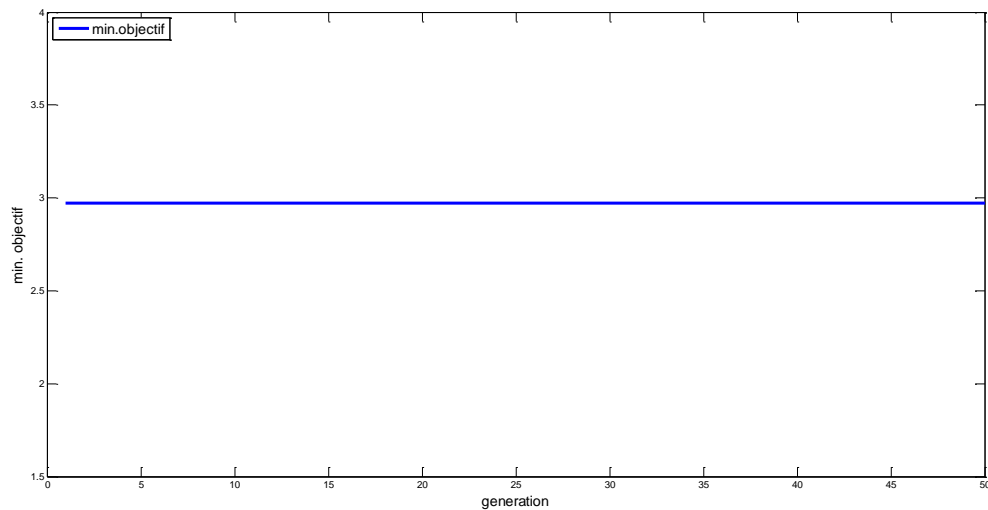


Figure IV.4: Valeur optimale de la fonction objective.

Chapitre IV : Recherche sur les algorithmes PSO (particle swarm optimization), simulation et interprétation du résultat

○ Sur la figure IV.4 est donnée la valeur minimale, optimale de la fonction objective environ 3.

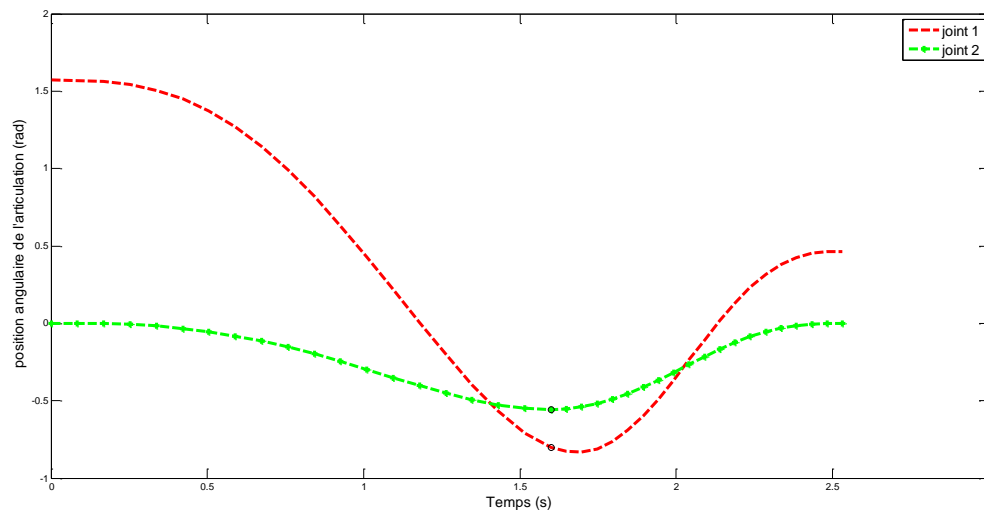


Figure IV.5: Trajectories de positions angulaires articulaires.

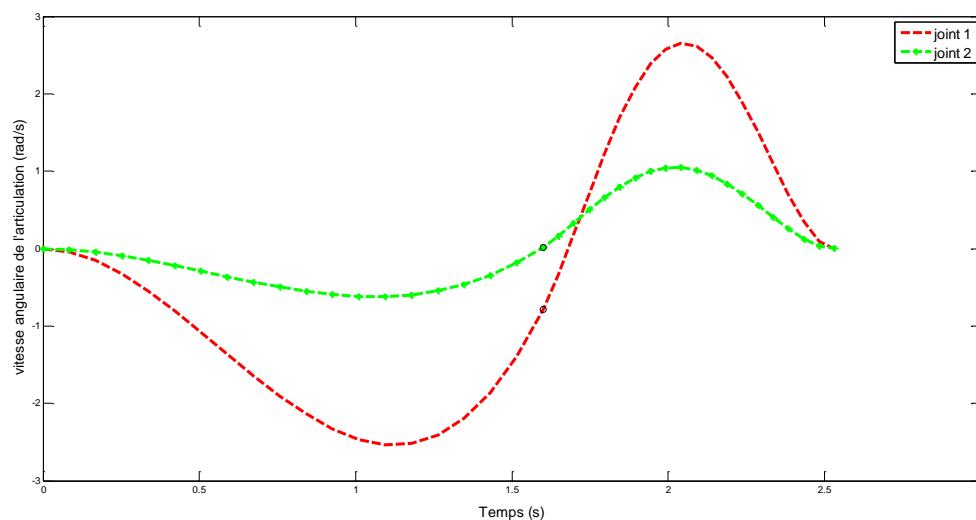


Figure IV.6 : Trajectories de vitesses angulaires articulaires.

○ Les figures IV.5, IV.6 IV.7 montrent successivement les trajectoires optimales des positions, vitesses et accélérations articulaires, elles passent par le point intermédiaire vers le point final selon le profil des splines.

Chapitre IV : Recherche sur les algorithmes PSO (particle swarm optimization), simulation et interprétation du résultat

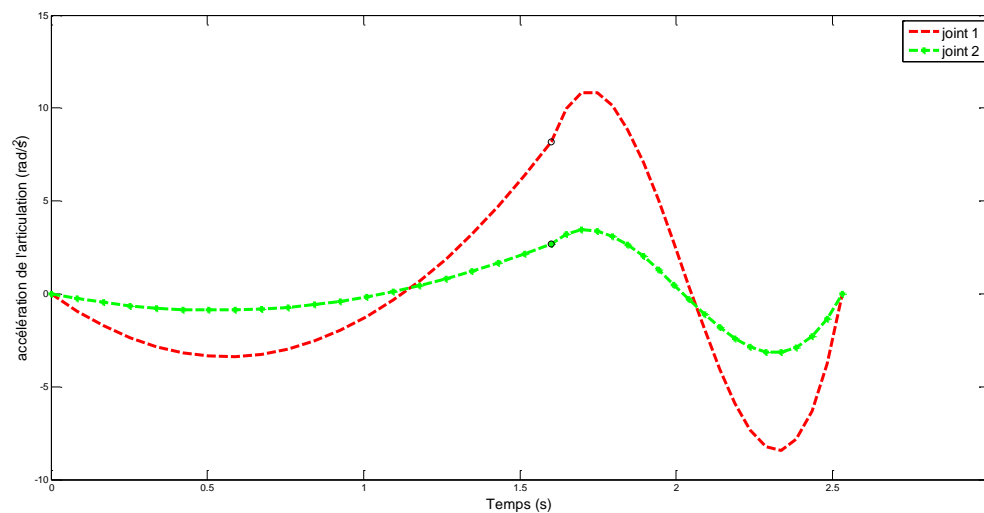


Figure IV.7 : Trajectoires d'accélération angulaires articulaires.

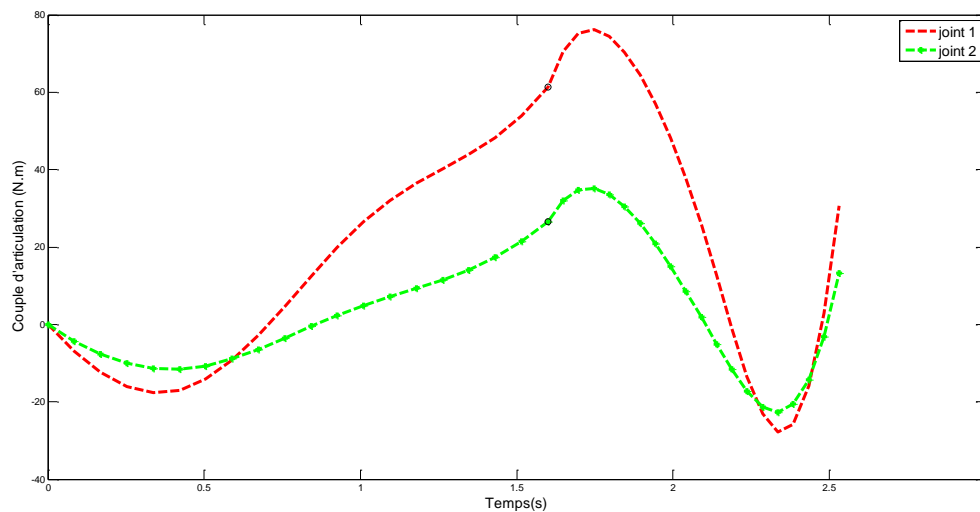


Figure IV.8 : Couples articulaires.

○ La figure IV.8 présente les couples optimaux développés par les actionneurs, nous remarquerons qu'ils dépassent de loin les valeurs limites de $\tau_{1\max} = 34 \text{ N}$ et $\tau_{2\max} = 12 \text{ N}$ prédéterminés.

Chapitre IV : Recherche sur les algorithmes PSO (particule swarm optimization), simulation et interprétation du résultat

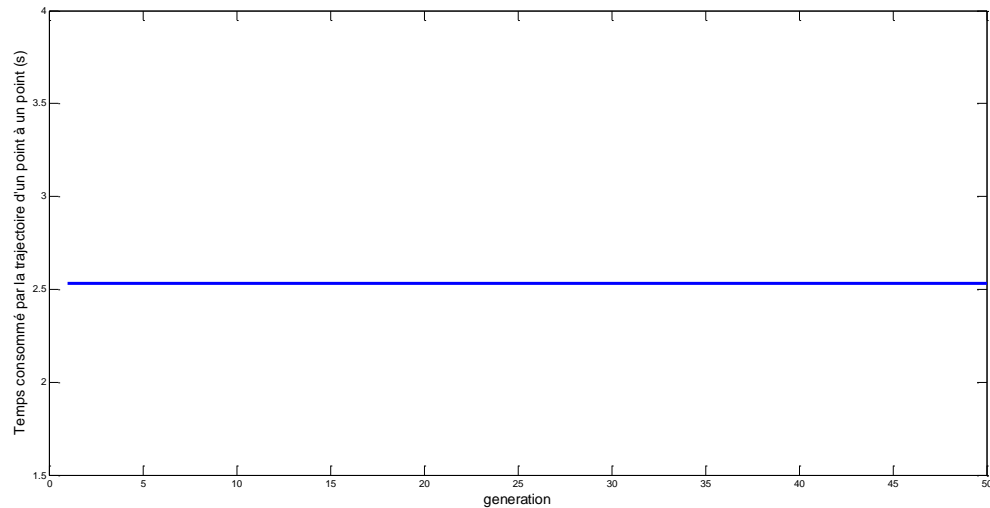


Figure IV.9 : Temps consommé par les trajectoires d'un point à un autre.

- Sur la figure IV.9 est donné le temps optimal total d'exécution de la trajectoire t_1+t_2 qui est environ de 2.5 secondes, c'est un temps relativement court vue la taille du robot considéré.

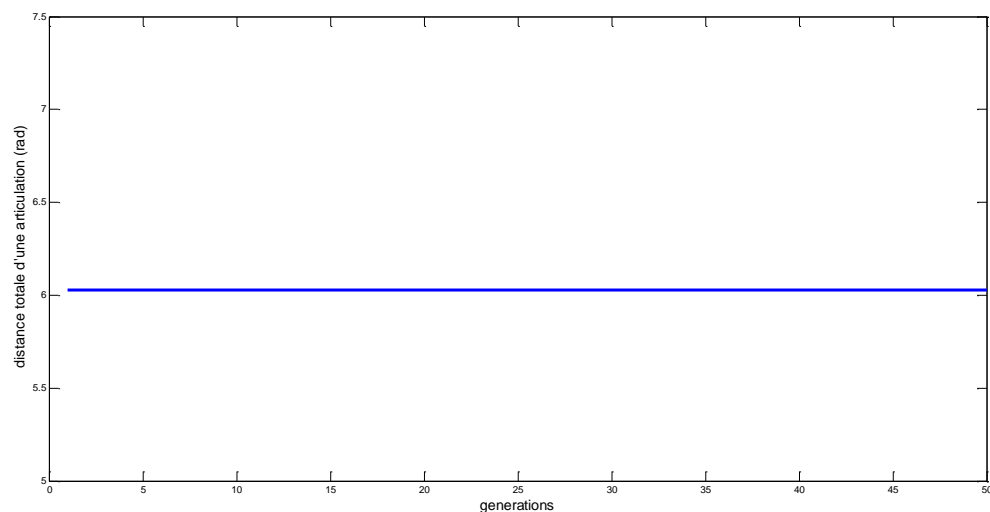


Figure IV.10 : Distance totale parcourue par une articulation.

- La figure IV.10 nous indique la distance optimale totale parcourue par les articulations du robot qui est de 6.1 rad et la figure IV.11 nous indique la longueur cartésienne optimale totale parcourue par le robot qui est de 10.52 m.

Chapitre IV : Recherche sur les algorithmes PSO (particle swarm optimization), simulation et interprétation du résultat

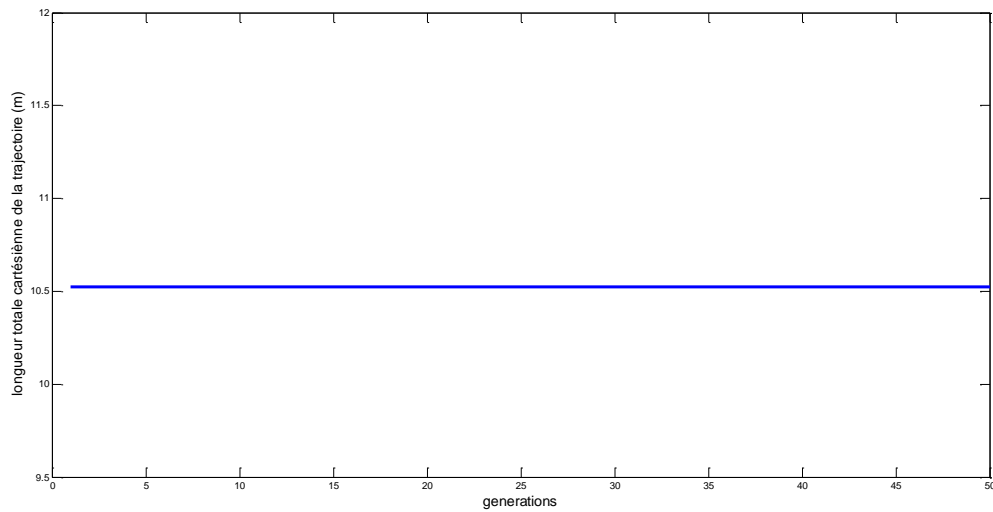


Figure IV.11: Longueur totale cartésienne de la trajectoire.

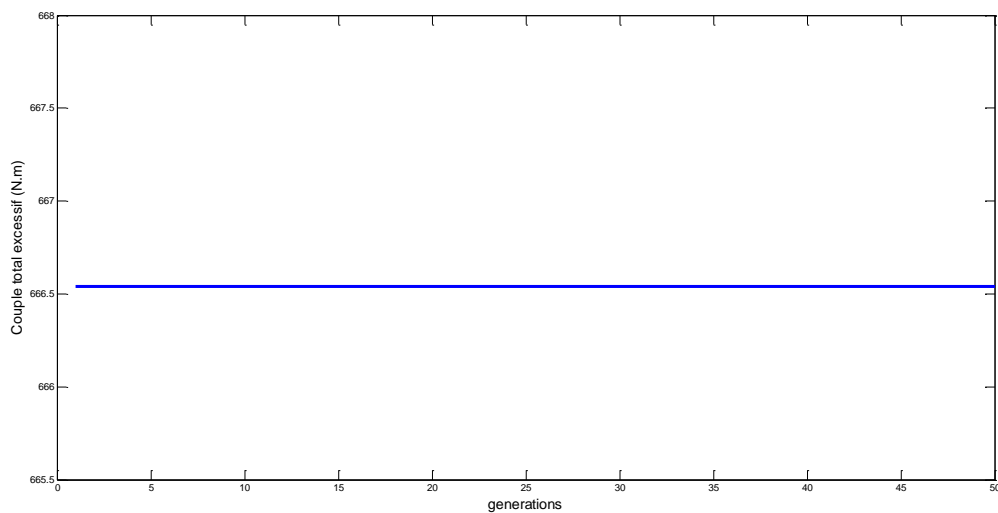


Figure IV.12 : Couple total excessif.

○ Enfin la figure IV.12 révèle la valeur du couple total excessif développé par le robot en exécutant la trajectoire qui est d'environ 667 Nm, cette valeur est relativement faible ce qui indique une optimisation de l'énergie consommée par le robot pendant son déplacement.

Chapitre IV : Recherche sur les algorithmes PSO (particule swarm optimization), simulation et interprétation du résultat

➤ Ces derniers résultats n'étant pas satisfaisants, notamment en termes de valeur de fonction objective qui est relativement élevé et les contraintes de couples violées, alors un autre essai en variant les paramètres de réglage de l'algorithme du PSO s'impose.

2^{ème} Essai :

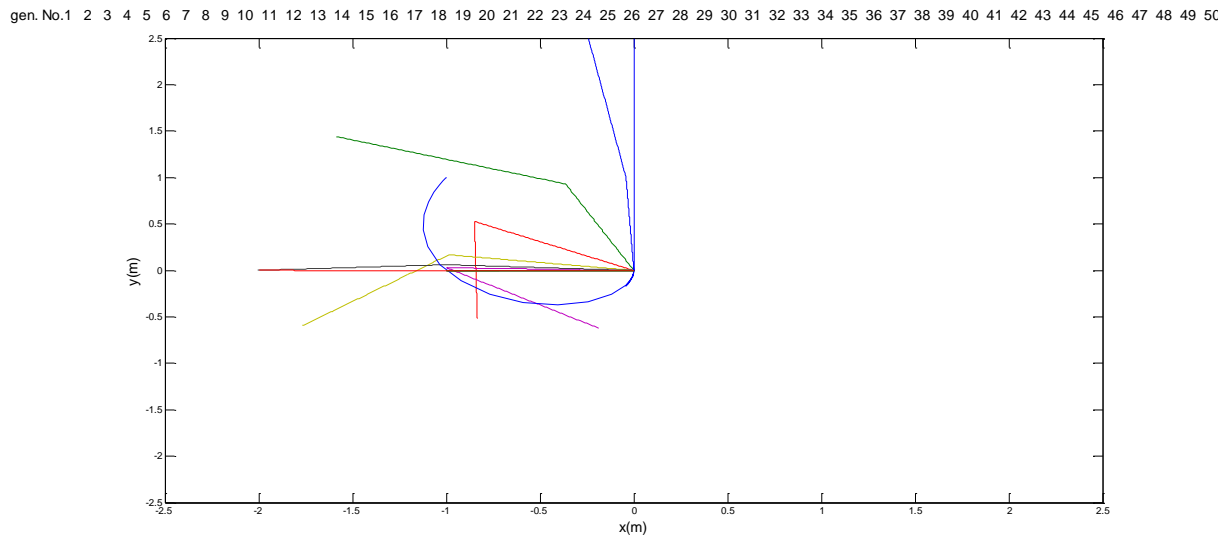


Figure IV.13 : Trajectoire du robot dans l'espace cartésien.

○ La figure IV.13 montre la trajectoire cartésienne optimale du robot planaire reliant les points initial et final dans l'espace opérationnel,

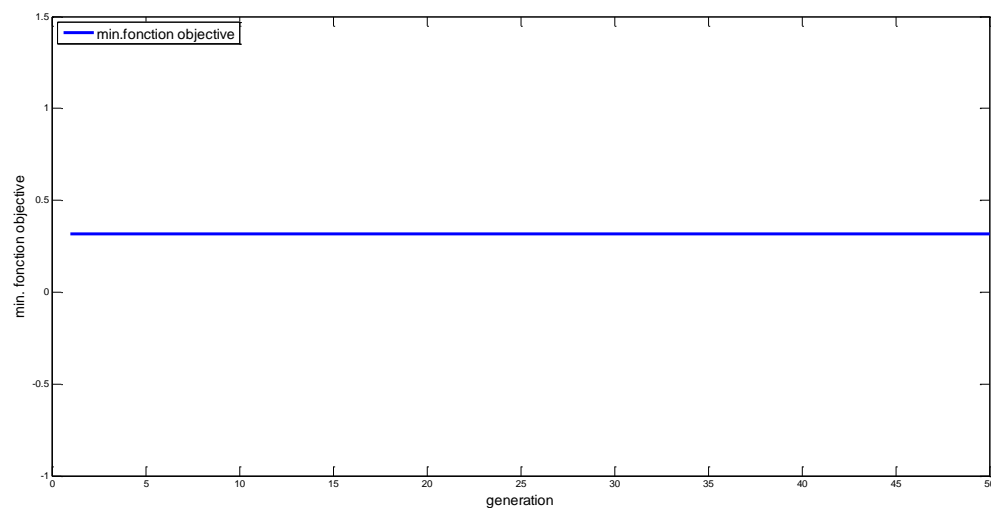


Figure IV.14 : Valeur optimale de la fonction objective.

Chapitre IV : Recherche sur les algorithmes PSO (particle swarm optimization), simulation et interprétation du résultat

○ Sur la figure IV.14 est donnée la valeur minimale, optimale de la fonction objective elle est inférieure à 1, soit environ 0.4.

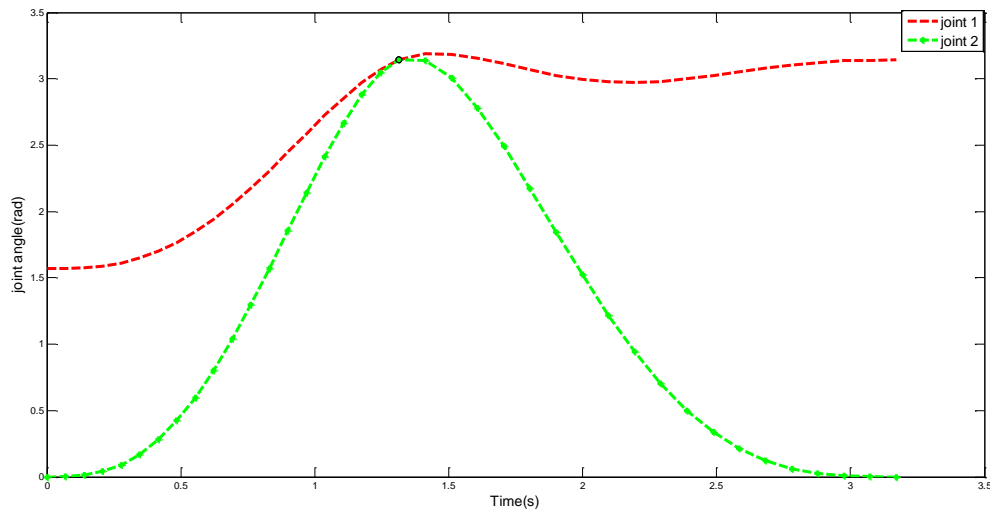


Figure IV.15 : Trajectoires de positions angulaires articulaires.

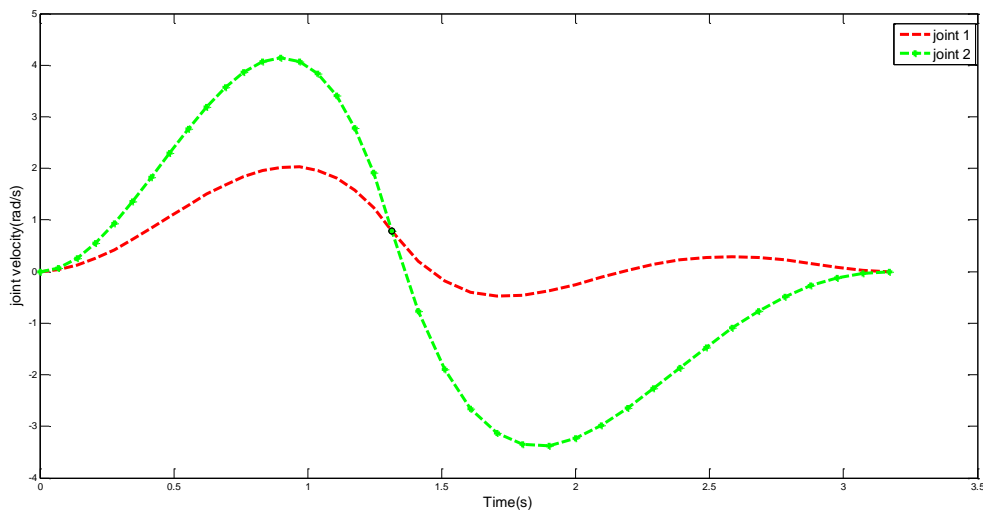


Figure IV.16 : Trajectoires de vitesses angulaires articulaires.

○ Les figures IV.15, IV.16 IV.17 montrent successivement les trajectoires optimales des positions, vitesses et accélérations articulaires, elles passent par le point intermédiaire vers le point final selon le profil des splines.

Chapitre IV : Recherche sur les algorithmes PSO (particle swarm optimization), simulation et interprétation du résultat

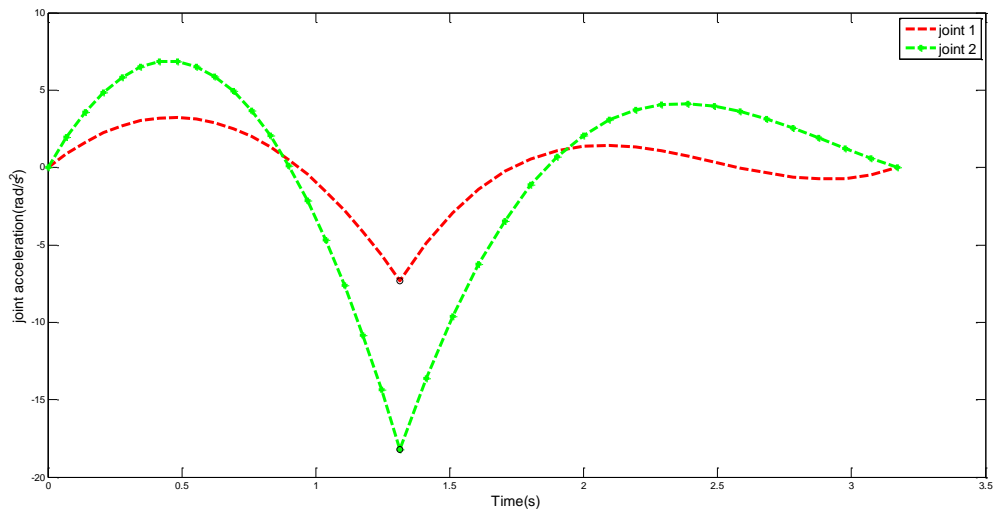


Figure IV.17 : Trajectoires d'accélération angulaires articulaires.

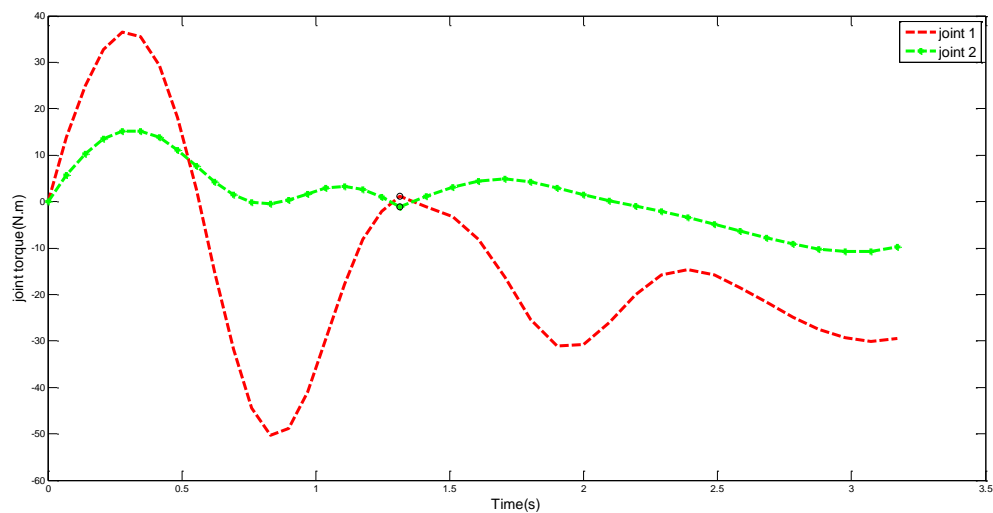


Figure IV.18 : Couples articulaires.

○ La figure IV.18 présente les couples optimaux développés par les actionneurs, nous remarquerons qu'ils ne dépassent pas les valeurs limites de $\tau_{1\max} = 34 \text{ N}$ et $\tau_{2\max} = 12 \text{ N}$ prédéterminés.

Chapitre IV : Recherche sur les algorithmes PSO (particle swarm optimization), simulation et interprétation du résultat

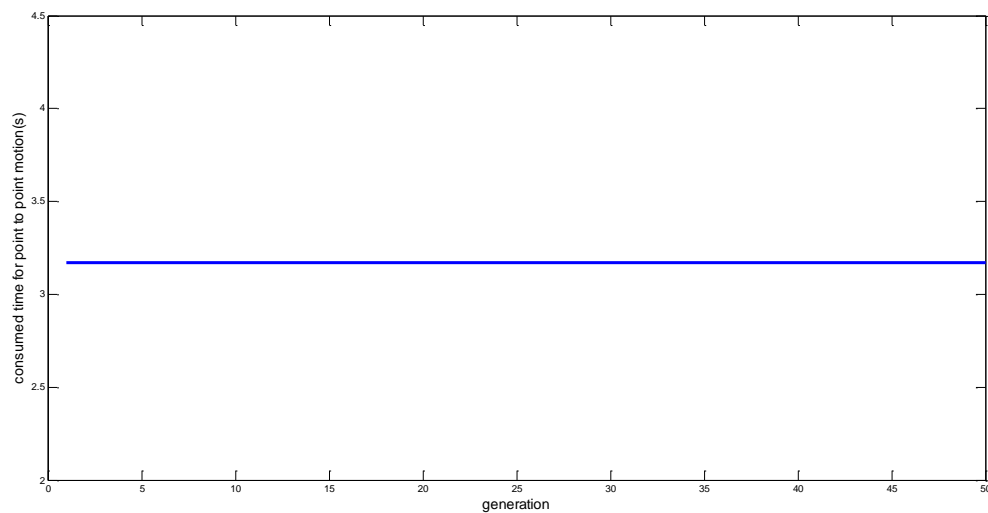


Figure IV.19 : Temps consommé par les trajectoires d'un point à un autre.

- Sur la figure IV.18 est donné le temps optimal total d'exécution de la trajectoire t_1+t_2 qui est environ de 3.25 secondes, c'est moins que le temps obtenu au 1^{er} essai.

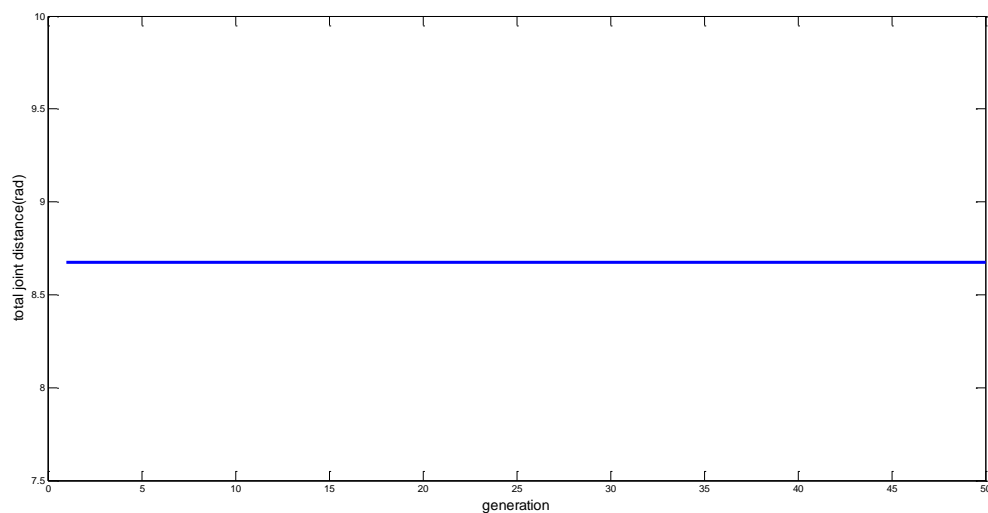


Figure IV.20 : Distance totale parcourue par une articulation.

Chapitre IV : Recherche sur les algorithmes PSO (particule swarm optimization), simulation et interprétation du résultat

○ La figure IV.20 nous indique la distance optimale totale parcourue par les articulations du robot qui est de 8.7 rad et la figure IV.21 nous indique la longueur cartésienne optimale totale parcourue par le robot qui est de 2.75 m, soit beaucoup plus courte que celle du 1^{er} essai.

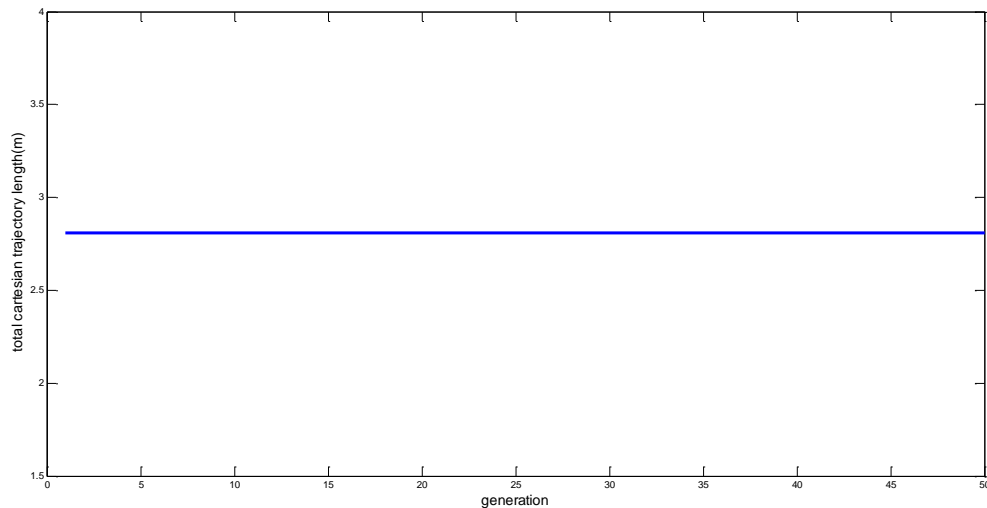


Figure IV.21 : Longueur totale cartésienne de la trajectoire.

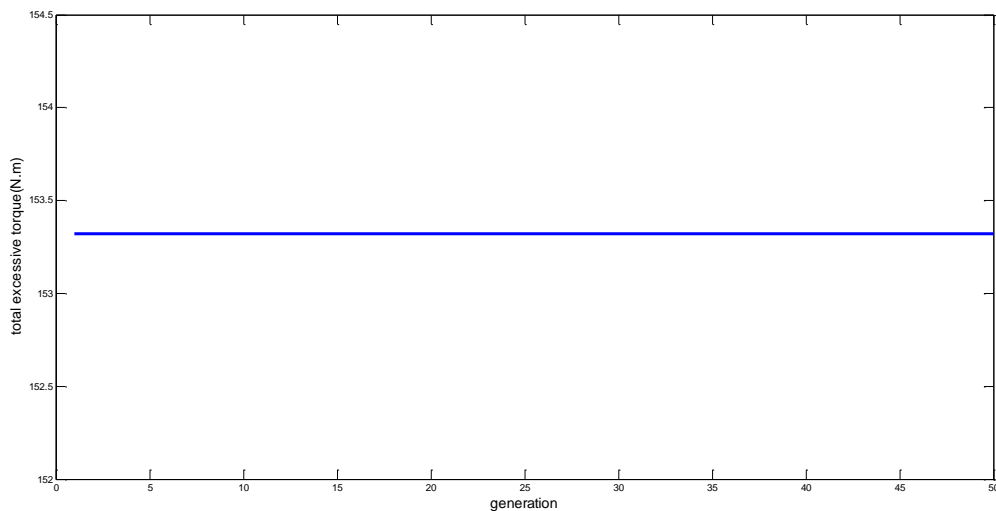


Figure IV.22 : Couple total excessif.

○ Enfin la figure IV.22 révèle la valeur du couple total excessif développé par le robot en exécutant la trajectoire qui est d'environ 153.35 Nm, cette valeur est relativement beaucoup plus

Chapitre IV : Recherche sur les algorithmes PSO (particule swarm optimization), simulation et interprétation du résultat

Faible que celle du 1^{er} essai, ce qui indique une optimisation de l'énergie consommée par le robot pendant son déplacement.

➤ Ces derniers résultats étant plus satisfaisants, que celle obtenues au 1^{er} essai, notamment au niveau des contraintes de couples non violées, alors cet essai est validé comme donnant les résultats satisfaisants et admissibles.

IV.6 Conclusion :

Les résultats obtenus dans ce travail confirment l'efficacité de la technique d'optimisation par essaim des particules, les trajectoires obtenues possèdent les profils optimaux en position, vitesse et accélération, le temps consommé est minimal ainsi que l'énergie représentée sous forme de couples optimaux d'actionneurs, les distances parcourues par les articulations et la longueur totale cartésienne sont aussi minimisées.

Deux essais ont été réalisés, le second essai semble donner les résultats les plus satisfaisants, dépassant de loin ceux donnés par le premier essai.

Les méthodes analytiques telles que la méthode du gradient ou celle de Newton donneront des résultats moins performants à cause des contraintes présentes dans le problème d'optimisation.

Conclusion générale

Conclusion générale et perspectives :

Les robots manipulateurs représentent un champ d'application et de développement très actif actuellement et dans un avenir proche, avec l'adjonction de l'intelligence artificielle ils incarneront l'industrie 2.0 du présent et du futur, l'architecture planaire d'un robot manipulateur est aussi importante et intéressante que l'architecture spatiale, malgré la simplicité de leur modèle dynamique, géométrique et cinématique, la génération de trajectoires optimales demeure un challenge et nécessite un investissement théorique et pratique. Ce domaine de recherches a connu beaucoup de contributions dans la littérature, avec surtout des techniques mathématiques classiques, tels les algorithmes de descente de gradient ou de Newton, mais pour les techniques heuristiques, l'intérêt ne s'est manifesté que récemment. C'est dans cette optique que s'inscrit ce travail comme une contribution dans cet axe.

L'apport des techniques d'optimisation dans l'amélioration des performances de fonctionnement des systèmes dynamiques utilisés dans la technologie moderne n'est pas à discuter, reste à les maîtriser dans les applications et se donner les moyens nécessaires pour y arriver, tel des calculateurs puissants et des algorithmes bien établis et robustes.

Notre travail a été concerné par l'optimisation des trajectoires de mouvements d'un bras manipulateur planaire à 2ddl, l'algorithme choisi était l'essaim de particules PSO, des travaux similaires déjà effectués utilisaient les algorithmes génétiques, par la nature heuristique de ces algorithmes les résultats obtenus semblent prometteurs, mais à un prix de temps de calcul et de place de mémoire exorbitant, néanmoins l'efficacité est au rendez-vous. Les paramètres de réglage de l'algorithme PSO jouent un rôle prépondérant dans la qualité des résultats, citant les par ordre d'importance : le choix du nombre d'individus, le nombre maximal d'itérations et les facteurs de pondération dans la fonction objective. C'est cette dernière qui a été le facteur décisif dans la qualité des derniers résultats performants,

Comme perspectives nous proposons l'optimisation par les techniques analytiques afin d'établir une comparaison entre les résultats des deux techniques et confirmer ou affirmer l'efficacité de l'algorithme du PSO.

Bibliographie

➤ Bibliographie :

- [1] F. Z.HAMDACHE, « Réalisation et commande d'un bras robotique à quatre degrés de liberté ».Mémoire de Master de l'UNIVERSITE AKLI MOHAND OULHADJE-, Bouira.
- [2] Mémoire de Master "Etude et réalisation d'une main robotique" ,Université mouloud mammeri , TIZI-OUZOU, 2018.
- [3] B. YAHYAOUIR, "ETUDE ET REALISATION D'une Main robotique",Mémoire de Master de l'UNIVERSITE MOULOU D MAMMERI DE TIZI-OUZOU, TIZI-OUZOU, juillet 2018.
- [4] A. A.Djebarni, "Etude et conception d'un bras de robot",Mémoire de MASTER de l'Université Mohamed El Bachir El Ibrahimi Bordj Bou Arréridj, Bordj Bou Arréridj, 2020.
- [5] I. Belkadi.H, "Commande d'un bras manipulateur a deux degrés de liberté suspendu",Mémoire de Master de l'UNIVERSITE ABDERAHMANE MIRA DE BEJAIA, BEJAIA, 2021-2022.
- [6] D.HAMMOUD « Apprentissage Automatique dans un Agent » Thèse de Doctorat de l'Université Mentouri Constantine, Algérie.
- [7] B. Mahesh, «Machine Learning Algorithms,» International Journal of Science and Research (IJSR), vol. Volume 9, n° %11, pp. 381-383, January 2020.
- [8] D.SAGNA, « Modélisation des précipitations par l'intelligence artificielle ou Apprentissage automatique en Casamance» Mémoire de Master Physique et Applications de l'Université Assane Seck de Ziguinchor, 27 Mars 2021.
- [9] E.GAUTHIER, « Utilisation des réseaux de neurones artificiels pour la commande d'un véhicule autonome », Thèse de Doctorat de l'institut national polytechnique de Grenoble,France, janvier 1999.
- [10] W. P. W. S. MCCULLOCH, «« A logical calculus of ideas immanent in nervous activity»,» *Bulletin of Mathematical Biophysics*,, vol. 5, pp. 115-133, 1943.
- [11] A.M.ARCHER, « Contrôle direct du couple électromagnétique des machines asynchrones de grande puissance», Thèse de Doctorat de l'institut national polytechnique de Toulouse, France, 1999.
- [12] «Wikipédia,» 2 Août 2023. [En ligne]. Available: <http://fr.m.wikipedia.org/wiki/Neurone>.

- G. BUREL, « Réseaux de neurones en traitement d'images: des modèles théoriques aux applications industrielles », Thèse de Doctorat de l'université de Bretagne Occidentale, France , 1999.
- [13] E. BENYOUSSEF, « Commande directe du couple flou et neuronale d'un moteur synchrone à double étoile sans capteur mécanique alimenté par onduleurs multiniveaux », Thèse de Doctorat de l'université de Sidi Bel Abbes, Algérie, 2015.
- [14] M. BENYAHIA, « Approche neuro-génétique pour le problème du dispatching économique environnemental », Thèse de Doctorat de l'université des sciences et de la technologie d'Oran, Algérie, Juillet 2012..
- [15] S. Mrs BELHADJER.H, "Classification des images avec les réseaux de neurones Convolutionnels",Mémoire de Master de l'Université Mouloud MAMMARI de Tizi-Ouzou, Tizi-Ouzou, 2017-2018.
- [16] P. CHEYNET, « Etude de la robustesse du contrôle intelligent face aux fautes induites par les radiations », Thèse de Doctorat de l'institut national polytechnique de Grenoble, France, Mai 1999.
- [17] R. GHAYOULA, « Contribution à l'Optimisation de la Synthèse des Antennes Intelligentes par les Réseaux de Neurones », Thèse de Doctorat de l'université de Tunis El Manar, Tunisie, Décembre 2008..
- [18] M. SAHRAOUI, « Contrôle robuste des systèmes non linéaires par les approches de l'intelligence artificielle », Thèse de Doctorat de l'université d'Oran1, Algérie, 2016.
- [19] «TP5 : Perceptron Master 2 SE Data Mining de l'Université Paul Sabatier,» [En ligne]. Available: <http://www.math.univ-toulouse.fr/~xgendre/ens/m2se/tp5.R>.
- [20] M. G. Gelly, "Réseaux de neurones récurrents pour le traitement automatique de la parole",Thèse de doctorat de l'Université Paris-Saclay, France, le 22 septembre 2017..
- [21] K. LAROUSSE, « Développement d'un superviseur flou appliqué à la commande d'un moteur à induction », Thèse de Doctorat de l'université de Boumerdes, 2016, Algérie.
- [22] R. Sadouni, « Commande par mode glissant flou d'une machine asynchrone à double étoile »,Mémoire de Magister de l'université de Sidi Bel Abbes, Algérie, 2010.
- [23] Z. L. BOUDJEMAA, « Commande vectorielle sans capteur de vitesse de la machine asynchrone à double alimentation par régulateur PI-flou », Mémoire de magister de l'école normale supérieure de l'enseignement technologique (ENSET) d'Oran, Algérie, 2009.

- [25] A.MEDJGHOU, « Application des techniques de l'intelligence artificielle à la commande des systèmes robotiques ». these de Doctorat de l'Université Batna 2 – Mostefa Ben Boulaïd, Batna, 08 Juillet 2018.
- [26] E. H. M. Assilian, «“An experiment in linguistic synthesis with a fuzzy”,» *Int. J. Man Mach. Studies*, vol. 7, n° %11, pp. 1-13, 1975.
- [27] B. Bouchon-Meunier, *La logique floue et ses applications*. Addison Wesley, Paris, 1995.
- [28] M. S. T. Takagi, «"Fuzzy identification of systems and its applications to modelling,» *IEEE trans. syst., man, cybern*, Vols. %1 sur %2SMC-15., n° %11, 1985.
- [29] A. M.BENNAI, "Localisation des stations mobile par la fusion",Mémoire de Master de l'Université Akli Mohand Oulhadj, bouira, september 2018.
- [30] CHAAL.M, "Modélisation cinématique d'un robot manipulateur à chaine continue ouverte",Mémoire de MASTER de l'UNIVERSITE KASDI MERBAH OUARGLA, OUARGLA,Algérie, juin 2013.
- [31] Bahloul.A, « Sur la commande des robots manipulateurs industriels en co-manipulation robotique », Mémoire présenté en vue de l'obtention du diplôme de Doctorat,Université Paris-Saclay, 2018.
- [32] AZOUIN, « Commande non linéaire d'un bras manipulateur »,Mémoire présenté en vue de l'obtention du diplôme de magister en électronique, 2009.
- [33] S.CARO, « CONCEPTION ROBUSTE DE MECANISMES »,Thèse de DOCTORAT de l'École Centrale de Nantes et l'Université de Nantes, Nantes , 17 décembre 2004.
- [34] SEBBAK.A, "Etude cinématique d'une machine outil à structure parallèle",Mémoire de Master de l'UNIVERSITE KASDI MERBAH OUARGLA, juin 2013.
- [35] M.CHAAL, « Modélisation cinématique d'un robot manipulateur à chaine continue ouverte »,Mémoire de mester de l'UNIVERSITE KASDI MERBAH OUARGLA, OUARGLA, 27 juin 2013.
- [36] R. Gourdeau, "Robotique : Modélisation des Robots Manipulateurs,Département de génie électrique, Ecole Polytechnique de Montréal, 4 novembre 2010..
- [37] A. Ghoul, Chapitre I Modélisation des robots manipulateurs., 2016.
- [38] D. L. KHENFAR.R, "Étude et réalisation d'un bras",MÉMOIRE DE Master de l'Université Ibn Khaldoun de Tiaret, Tiaret, 2021.
- [39] S.Maakaroun, Modélisation et simulation dynamique d'un véhicule urbain innovant en utilisant le formalisme de la robotique. Thèse de doctorat ,école des Mines de de l'Université Nante, france, Décembre 2011.

- [40] J.-L. Boimond, «« Cours de robotique » ,ISTIA Université Angers,,» France, p. 23.
- [41] D. E. Khalil.W, « Modélisation identification et commande des robots » ,2ème édition,Hermes Science Publication, Paris, 1999.
- [42] M. H.BELLOUATI, "Modélisation géométrique d'un bras robotique",Mémoire de Master de l'Université-Ain-Temouchent- Belhadj Bouchaib, Ain-Temouchent-, 2021.
- [43] M. s. Achour.H, "conception et réalisation d'un bras manipulateur a 3 degré de liberté",Mémoire de Master de l'université saad, blida, 2017.
- [44] KERRACIA, "COMMANDES ROBUSTES DES ROBOTS MANIPULATEURS RIGIDES "MAGISTER de l'Université d'Oran Es-Sénia, oran, Décembre 2004.
- [45] A.BOUREBBOU, "MODELISATION ET OPTIMISATION DES PERFORMANCES DYNAMIQUES D'UNE MACHINE OUTIL A STRUCTURE PARALLELE",Thèse de Doctorat de l'Université Mohamed khider, Biskra, septembre 2019.
- [46] B. AOUCHENNI.M, "Réalisation d'un environnement de modélisation, commande et simulation virtuel des robots manipulateurs",Mémoire de Master de l'Université A. MIRA – BEJAIA, BEJAIA, 2020.
- [47] LOUCIF.F, "Contribution à la commande non linéaire des robots manipulateurs",THÈSE De DOCTORAT de l'Université 8 Mai 1945 Guelma, Guelma, 2020-2021.
- [48] F.MAKHLOUFI, "Modélisation et commande des robots manipulateurs par les outils de l'intelligence artificielle",THESE De DOCTORAT de l'UNIVERCITE BADJI-MOKHAR-ANNABA, ANNABA.
- [49] Belarbi.S, "Commande et identification d'un bras de robot à deux degrés de liberté par les réseaux de neurones",Mémoire de Master de l'Université de Bejaia, Bejaia, juin 2013.
- [50] Mr:Belarbi.S, "Commande et identification d'un bras de robot à deux degrés de liberté par les réseaux de neurones",MÉMOIRE DE MASTER de l'Université de Bejaia, Bejaia, juin 2013.
- [51] M. DEFOORT, “Contributions a la planification et a la commande pour les robots mobiles coopératifs”,thèse de Doctorat, Ecole centrale de Lille, 2007.
- [52] I.H.Aguilar, "Commande des bras manipulateurs et retour visuel pour des applications à la robotique de service",THESE de Doctorat de l'Universit'e Toulouse III, septembre 2007, france.

- [53] N.ZERROUKI, Contribution à la planification des trajectoires par l'utilisation des métaheuristiques pour la commande des robots manipulateurs",Thèse de Doctorat de l'Université de Batna 2 – Mostefa Ben Boulaïd, batna.
- [54] BAZEM.B, "Planification de trajectoire à temps optimal d'un robot manipulateur à trois degrés de libertés",Mémoire Mémoire de fin d'étude de l'UNIVERSITE DE ABDERRAHMANE MIRA-BEJAIA., 2012.
- [55] K. L. H. TAZEBINTE, "Planification et optimisation de trajectoire d'un robot mobile par les algorithmes évolutionnaires",Mémoire de Master de l'ECOLE NATIONALE POLYTECHNIQUE, Hassen Badi, El-Harrach, Algérie, juin 2012.
- [56] B. SMAOUN.R, "Planification de trajectoire sans collision d'un robot manipulateur en utilisant l'approche de champs de potentiels artificiels",Mémoire de Mémoire de fin d'études de l'Université Abderrahmane Mira Bejaïa, 2018.
- [57] BOUNOUARA.N, "Planification de trajectoires pour les bras Manipulateurs",Mémoire de MAGISTER de l'UNIVERSITE DE BATNA, 2010.
- [58] Mouhaman.A, «Chapitre II : Méthodes de calcul des réseaux maillés de distribution d'eau,» pp. 41-42.
- [59] Kasmi.B, "PLANIFICATION DE TRAJECTOIRE D'UN ROBOT MOBILE"Mémoire de magister dr l'UNIVERSITÉ FERHAT ABBAS -SÈTIFUFAS, ALGERIE, juin 2012.
- [60] Mahideb.H, "L'optimisation multi-objectif basée sur l'intelligence computationnelle",Mémoire de Magister de l'Université Constantine 2, novembre 2014.
- [61] LOUCIF.F, "contribution à la commande non linéaire des robots manipulateurs",thèse de doctorat de l'université 8 mai 1945 Guelma, 2020.
- [62] E. Dombre, "Analyse et Modélisation des Robots Manipulateurs", 22 janvier 2002..

