

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE
SCIENTIFIQUE

UNIVERSITE FRERES MENTOURI CONSTANTINE 1
FACULTE DES SCIENCES DE LA TECHNOLOGIE
DEPARTEMENT DE GENIE MECANIQUE

N° de série :02/DS/2024

N° d'ordre :01/GM/2024

THESE

Présentée pour obtenir le diplôme de

DOCTORAT EN SCIENCES

Spécialité : GENIE MECANIQUE

Option : CONSTRUCTION MECANIQUE

Par :

MERRAD Ammar

THEME

CONTRIBUTION A LA MODELISATION ET A LA COMMANDE DES ROBOTS CONTINUUM

Soutenue le : 15 / 02 / 2024

Devant le Jury :

Président :	Mr. Ali BOUCHOUCHA	Prof.	Université de Constantine 1
Rapporteur :	Mr. Ammar AMOURI	MCA.	Université de Constantine 1
Examineurs :	Mr. Smail BENISSAAD	Prof.	Université de Constantine 1
	Mr. Ridha KELAIAIA	Prof.	Université de Skikda
	Mr. Allaoua BOULAHIA	MCA.	Université d'Oum el Bouaghi
Invité :	Mr. Abdelhakim CHERFIA	Prof.	Université de Constantine 1

Année Universitaire : 2023/2024

Remerciements

Mes remerciements vont tout d'abord à Dieu le tout-puissant pour la force et la volonté qui m'a procuré pour terminer mon travail.

Tout d'abord, je tiens à remercier vivement mon Directeur de Thèse : Monsieur **Ammar AMOURI**, Maître de Conférences à l'université des Frères Mentouri - Constantine 1, pour avoir encadré et dirigé ce travail, pour la confiance qu'il m'a accordée, sa patience, et ses conseils, ses orientations et le soutien que j'ai trouvé auprès de lui au long de la réalisation de ce travail.

Je remercie vivement Monsieur **Ali BOUCHOUCHA**, Professeur à l'université des Frères Mentouri - Constantine 1 pour l'honneur qui m'a été fait de présider le jury de cette thèse.

J'exprime également ma gratitude aux membres du jury qui m'ont honoré de leur participation:

Monsieur **Smail BENISSAAD**, Professeur à l'université de Constantine 1

Monsieur **Ridha KELAIAIA**, Professeur à l'université de Skikda

Monsieur **Allaoua BOULAHIA**, Maître de Conférences à l'université d'Oum el Bouahgi

Monsieur **Abdelhakim CHERFIA**, Professeur à l'université de Constantine 1

Je profite de cette occasion qui m'est offerte pour remercier mon épouse qui m'est beaucoup aidé et encouragé durant cette période.

Je tiens également à remercier tous ceux qui de près ou de loin ont participé à l'élaboration de ce travail.

Dédicaces

A mes parents,

A tout ma famille,

A tous mes amis,

Résumé

Ces dernières années, les robots continuum ont suscité beaucoup d'attention de la part de la communauté robotique en raison des caractéristiques et fonctionnalités inhabituelles qu'ils peuvent offrir par rapport à leurs homologues à liaisons rigides. Ces caractéristiques extraordinaires, à savoir : la flexibilité, la dextérité et la légèreté de structure, font les répandre rapidement de jour en jour dans les secteurs industriels ; cependant, leur modélisation et commande posent encore des défis. A cette fin, les deux problèmes sus-cités ont été abordés dans ce travail. Dans la première partie consacrée à la modélisation, un algorithme fiable a été proposé pour résoudre l'un des problèmes commun des robots continuum, à savoir : le modèle géométrique inverse et sa redondance. L'algorithme proposé se compose de deux sous-algorithmes. Le premier sous-algorithme est dédié pour déterminer toutes les solutions possibles du modèle géométrique inverse, dans lequel la première solution aléatoire est calculée à l'aide d'une méthode d'optimisation, à savoir : optimisation par essaim de particules, tandis que le second est consacré à la résolution du problème de redondance. Dans la deuxième partie, quatre contrôleurs ont été implémentés, à savoir le contrôleur : Proportionnel-Intégral Dérivé (PID), PID Optimisé (OPID), PID d'Ordre Fractionnaire (FOPID) et la commande prédictive non linéaires (NMPC), dans lesquels leur conception a été réalisée sur la base des modèles cinématique et dynamique. Enfin, l'efficacité de l'algorithme proposé est vérifiée par une série d'exemples de simulation. D'autre part, les quatre contrôleurs ont été évalués et comparés en termes de précision de suivi et de temps de calcul. Pour conclure, des études comparatives sont effectuées.

Mots-clés : Robot continuum, robot continuum actionné par câbles, modélisation géométriques, résolution de redondance, la commande, simulation.

Abstract

In recent years, continuum robots have received a lot of attention from the robotics community due to the unusual features and functionality they can offer compared to their rigid-linked counterparts. These extraordinary characteristics, namely: flexibility, dexterity and lightness of structure, make them spread rapidly day by day in the industrial sectors; however, their modeling and control still pose challenges. To this end, the two above-mentioned problems have been addressed in this work. In the first part dedicated to modeling, a reliable algorithm has been proposed to solve one of the common problems of continuum robots, namely: the inverse kinematic model and its redundancy. The proposed algorithm consists of two sub-algorithms. The first sub-algorithm is dedicated to determine all the possible solutions of the inverse kinematic model, in which the first random solution is calculated using an optimization method, namely: particle swarm optimization, while the second is devoted for solving the redundancy problem. In the second part, four controllers were implemented, namely the controller: Proportional-Integral-Derivative (PID), Optimized PID (OPID), Fractional Order PID (FOPID) and Nonlinear Model Predictive Control (NMPC), in which their design was carried out on the basis of kinematic and dynamic models. Finally, the efficiency of the proposed algorithm is verified by a series of simulation examples. On the other hand, the four controllers were evaluated and compared in terms of tracking accuracy and computation time. To conclude, comparative studies are performed.

Keywords: Continuum robot, cable-driven continuum robot, kinematics modeling, redundancy resolution, control, simulation.

ملخص

في السنوات الأخيرة ، حظيت الروبوتات المستمرة باهتمام كبير من طرف الباحثين نظرًا للسمات والوظائف الغير عادية التي يمكن أن تقدمها مقارنة بنظيراتها الصلبة. هذه الخصائص وهي: المرونة والبراعة وخفة الهيكل، تجعلها تنتشر بسرعة يومًا بعد يوم في القطاعات الصناعية ؛ ومع ذلك ، فإن نمذجتهم والتحكم فيها لا يزالان يشكلان تحديات. لهذه الغاية، تم تناول المشكلتين المذكورتين أعلاه في هذا العمل. في الجزء الأول المخصص للنمذجة ، تم اقتراح خوارزمية موثوقة لحل إحدى المشكلات الشائعة للروبوتات المتصلة وهي: النموذج الحركي العكسي وتكراره. تتكون الخوارزمية المقترحة من خوارزميتين فرعيتين. تم تخصيص الخوارزمية الفرعية الأولى لتحديد جميع الحلول الممكنة للنموذج الحركي العكسي، حيث يتم حساب الحل العشوائي الأول باستخدام طريقة التحسين وهي: طريقة تحسين سرب الجسيمات ، بينما تم تخصيص الخوارزمية الفرعية الثانية لحل مشكلة التكرار. في الجزء الثاني تم تنفيذ أربع وحدات تحكم وهي: المتحكم التناسبي التكاملي التفاضلي (PID)، PID المحسن (OPID)، PID ذي الرتبة الكسرية (FOPID) والتحكم التنبئي الغير خطي (NMPC)، حيث تم تنفيذ تصميمهم على أساس النماذج الحركية والديناميكية. أخيرًا، تم التحقق من كفاءة الخوارزمية المقترحة من خلال سلسلة من أمثلة المحاكاة. من ناحية أخرى، تم تقييم وحدات التحكم الأربعة ومقارنتها من حيث دقة التتبع ووقت الحساب. في الختام ، تم إجراء دراسات مقارنة.

كلمات مفتاحية : روبوت مستمر ، روبوت مستمر مشغل بكابلات ، نمذجة حركية ، تكرار ، تحكم ، محاكاة

Table des Matières

Table des Matières	iv
Liste des Figures	viii
Liste des Tableaux	xi
Nomenclature	xii
Acronymes	xv
Introduction générale	2
A. Motivation et contexte de la thèse	2
B. Objectifs de la thèse	2
C. Contributions	3
D. Résultats obtenus	3
E. Organisation du manuscrit	4
1. Etat de l’art sur les robots continuum	7
1.1. Introduction	7
1.2. Les robots continuum	7
1.3. Classification des robots continuum	11
1.3.1. Classification selon la structure du robot	11
1.3.2. Classification selon l’endroit des actionneurs	12
1.4. Applications et défis des robots continuum	13
1.5. Avantages et inconvénient des robots continuum	14
1.6. Approches et méthodes de modélisation	15
1.6.1. Modèles géométriques	15
1.6.2. Modèles dynamiques	17
1.7. Stratégies de commande	18
1.8. Conclusion	18

2. Modélisation géométrique	22
2.1. Introduction	22
2.2. Description géométrique de la structure du robot	22
2.3. Hypothèses de modélisation.....	24
2.4. Modélisation géométrique directe.....	25
2.4.1. Modèle géométrique direct d'une unité (j, k)	26
2.4.2. Modèle géométrique direct d'une section flexible k	30
2.4.3. Modèle géométrique direct d'un robot continuum multi-sections	30
2.5. Modélisation géométrique inverse.....	30
2.5.1. Modèle géométrique inverse d'une unité (j, k)	31
2.5.2. Modèle géométrique inverse d'une section k	32
2.5.3. Modèle géométrique inverse d'un robot continuum multi-sections	32
2.5.3.1. L'algorithme PSO	34
2.5.3.2. La fonction de coût	36
2.5.3.3. Sélection des paramètres de l'algorithme PSO	36
2.5.3.4. Condition aux limites	36
2.5.3.5. Limitations physiques	37
2.5.4. Exemples d'application	37
2.5.4.1. Application 1	37
2.5.4.2. Application 2	39
2.6. Simulations numériques	41
2.6.1. Détermination de l'espace de travail	42
2.6.1.1. Détermination de l'espace de travail d'un robot à une section flexible	42
2.6.1.2. Détermination de l'espace de travail d'un robot multi-sections..	44
2.6.2. Simulation des modèles géométrique inverse utilisant le sous- Algorithme 1	44
2.6.3. Contributions récentes sur la modélisation géométrique inverse des robots continuum	49
2.7. Conclusion.....	50
3. Résolution de la redondance du modèle géométrique inverse	53
3.1. Introduction.....	53

3.2.	Critères de résolution de la redondance du MGI d'un robot continuum	53
3.3.	Stratégie d'évitement d'obstacle	54
3.4.	Application	55
3.4.1.	Application 1 : utilisant le critère de minimisation du vecteur des variables des actionneurs	55
3.4.2.	Application 2 : utilisant le critère des distances parcourues par des extrémités des premières $(n - 1)$ sections flexibles	58
3.5.	Illustration numérique du schéma fonctionnel de l'algorithme proposé	61
3.6.	Simulations numériques	64
3.6.1.	Simulation des modèles géométriques inverses utilisant le sous-algorithme 2	64
3.6.2.	Simulation du modèle géométrique inverse en présence d'obstacle	67
3.6.3.	Contributions récentes sur la résolution de redondance du modèle géométrique inverse	69
3.7.	Conclusion.....	70
4.	Commande : PID, OPID, FOPID, NMPC	73
4.1.	Introduction	73
4.2.	Contrôleur PID	73
4.3.	Contrôleur OPID	75
4.4.	Contrôleur PID d'ordre fractionnaire.....	76
4.4.1.	Préliminaires.....	76
4.4.1.1.	L'intégrale fractionnaire de Riemann-Liouville	76
4.4.1.2.	La dérivée fractionnaire de Riemann-Liouville	77
4.4.1.3.	La dérivée fractionnaire de Grünwald-Letnikov	78
4.4.1.4.	La dérivée fractionnaire de Caputo	78
4.4.2.	Structure du contrôleur PID d'ordre fractionnaire	79
4.5.	La commande prédictive	80
4.5.1.	Formulation générale de la commande prédictive non linéaire	81
4.6.	Modèles mathématiques du robot continuum	82
4.6.1.	Positions et orientations	82
4.6.2.	Modèle cinématique	83
4.3.3.	Modèle dynamique	84

4.7.	Applications : implémentation des contrôleurs	85
4.7.1.	Application 1 : implémentation de PID, OPID et FOPID	86
4.7.1.1.	Exemple 1 : suivi d'un point de consigne	87
4.7.1.2.	Exemple 2 : suivi d'une trajectoire	89
4.7.2.	Application 2 : implémentation de NMPC	91
4.7.2.1.	Exemple 1 : suivi d'un point de consigne	92
4.7.2.2.	Exemple 2 : suivi d'une trajectoire	93
4.7.3.	Comparaison des stratégies de commande	96
4.8.	Conclusion	97
	Conclusion générale	89
	Annexe	101
	Références bibliographiques	103

Liste des Figures

1.1	Différenciation : liée et continue. (a) structure à liaisons discrète rigide, (b) structure serpentine, et (c) structure continue [13].....	8
1.2	Exemples de saisie d'objets : (a) Robot OctArm IV [15], (b) Robot Air-Octor [16], Octopus-Arm [17].....	8
1.3	Tensor Arm manipulator [18].....	9
1.4	Exemples d'actionnement extrinsèque : (a) robot actionné par câbles [37], (b) robot actionné par tendons [38].....	12
1.5	Exemples d'actionnement intrinsèque : (a) actionnement pneumatiques [39], (b) actionnement musculaire pneumatique artificiel (PMA) [40], et (c) PMA tressé [41]	12
1.6	Exemple d'actionnement extrinsèque et intrinsèque [42].....	13
1.7	Modèles mathématique de modélisation des robots.....	15
1.8	Méthodes et approches de résolution des modèles géométriques directs des robots continuum.....	16
2.1	Structure d'un robot continuum actionné par câbles à trois sections flexibles : (a) un robot continuum à trois sections flexibles, (b) une section du robot, et (c) une unité de la section flexible modélisée comme un robot parallèle de type 3UPS-1PU.....	23
2.2	Vue globale de la modélisation géométrique basée sur l'approche de l'hypothèse à courbure constante.....	26
2.3	Unité j de la section flexible k modélisée comme un arc de cercle dans l'espace.....	26
2.4	Déformation de l'unité (j, k) selon un arc de cercle: (a) Différentes longueurs et rayons de courbure de l'unité ($\varphi_k = 0$), (b) Différents rayons de courbure de l'unité ($\varphi_k \neq 0$).....	28

2.5	Vue globale du schéma fonctionnel de l’algorithme fournissant des solutions du MGI et la résolution de la redondance pour un robot continuum à n sections flexibles.....	33
2.6	Représentation d’un cercle passant par trois points quelconque dans l’espace.....	39
2.7	Espace de travail en 3D d’un robot continuum à une section flexible.....	43
2.8	Espace de travail de chaque section flexible d’un robot continuum à deux sections flexibles pour un point donné dans l’espace 3D.....	43
2.9	Espace de travail de chaque section flexible d’un robot continuum à trois sections flexibles pour un point donné dans l’espace 3D.....	44
2.10	Représentations bidimensionnelle d’un espace de travail d’un robot continuum à deux sections flexibles.....	45
2.11	Représentations bidimensionnelle d’un espace de travail d’un robot continuum à trois sections flexibles.....	45
2.12	Toutes les solutions possibles du MGI d’un robot continuum à deux sections flexibles pour une pose désirée dans l’espace.....	46
2.13	Configuration initiale du robot à trois sections flexibles.....	47
2.14	Les points finals atteignables de deux premières sections flexibles pour atteindre un point désiré de la trajectoire.....	47
2.15	Trois configurations aléatoires du robot pour atteindre un point donné de la trajectoire désirée.....	48
3.1	Stratégie d’évitement des obstacles.....	54
3.2	Génération des configurations optimales du robot à trois sections flexibles utilisant le critère de minimisation des longueurs de câbles.....	57
3.3	Génération des configurations optimales du robot à trois sections le critère de minimisation des distances parcourues par des extrémités des premières $(n - 1)$ sections flexible	60
3.4	Exemple numérique illustrant le schéma fonctionnel de l’algorithme proposé....	63
3.5	Configurations optimales du robot (a) 1er critère, (b) 2ième critère.....	65
3.6	Le profile des longueurs de câbles nécessaires pour suivre la trajectoire spirale pour les deux critères.....	65
3.7	Configurations optimales du robot à trois sections flexibles pour un suivi d’une trajectoire linéaire. (a) 1ier critère (minimisation du vecteur des variables des	

	actionneurs), (b) 2ième critère (minimisation des distances parcourues)	66
3.8	Les vecteurs des variables des actionneurs nécessaires pour suivre la trajectoire linéaire par les deux critères.....	66
3.9	Configurations de collision et d'évitement d'obstacle du robot (les points rouges sont des zones de collisions, et verts sont des zones libres) : (a) collision, (b) évitement d'obstacle.....	68
4.1	Structure du contrôleur PID.....	75
4.2	Structure de réglage des paramètres du contrôleur PID.....	75
4.3	Relation entre les contrôleurs PID et FOPID.....	79
4.4	Structure de réglage des paramètres du contrôleur FOPID par la méthode PSO ...	80
4.5	Structure de la commande NMPC avec la méthode PSO.....	81
4.6	Structure des commandes PID, OPID et FOPID en boucle fermée.....	87
4.7	Comparaison des réponses des contrôleurs PID, OPID et FOPID pour un suivi d'un point de consigne.....	87
4.8	Réponses du contrôleur FOPID avec $\mu = 1$ et différentes valeurs de λ	88
4.9	Réponses du contrôleur FOPID avec $\lambda = 1$ et différentes valeurs de μ	89
4.10	Signal de commande nécessaire pour un suivi d'une trajectoire.....	89
4.11	La trajectoire désirée et générée par FOPID et les erreurs Euclidiennes entre eux	90
4.12	Les angles de flexion désiré et généré et les erreurs entre eux	91
4.13	Résultats de simulation du point de consigne évaluant la convergence du contrôleur NMPC avec les deux modèles cinématique et dynamique.....	92
4.14	Résultats de simulation du point de consigne évaluant la convergence des deux contrôleurs NMPC associés aux modèles cinématique et dynamique.....	93
4.15	Les profils des trajectoires désirée et générée et les erreurs Euclidiens le long des axes x et y	94
4.16	Temps de la simulation du point de consigne pour évaluer la stabilisation du contrôleur NMPC avec les deux modèles cinématique et dynamique	95

Liste des Tableaux

1.1	Comparaison de quelques robots continuum avec leur bio-inspiration.....	9
1.2	Quelques types de robots continuum classés selon leur structure.....	11
1.3	Quelques fonctions d'applications des robots continuum.....	13
1.4	Quelques avantages et inconvénients des robots continuum.....	14
2.1	Paramètres géométriques du CDCR utilisé dans les simulations numériques.....	42
2.2	Quelques contributions traitant le problème du MGI des robots continuum.....	49
3.1	Poses optimales des extrémités de la première et la deuxième section flexible utilisant les deux critères de minimisation.....	64
3.2	Points optimaux des extrémités de la première et deuxième de section flexible satisfont les deux critères tout en évitant l'obstacle.....	69
3.3	Les longueurs de câbles nécessaires satisfont les deux critères tout en évitant l'obstacle.....	69
3.4	Quelques contributions traitant le problème de la redondance des robots continuum.....	69
4.1	Influence des paramètres du contrôleur PID.....	74
4.2	Paramètres du robot continuum [55].....	86
4.3	Les paramètres optimaux des contrôleurs PID, OPID et FOPID.....	87
4.4	MSE pour un suivi d'un point de consigne.....	88
4.5	Quelques contributions sur le contrôle des robots continuum.....	96

Nomenclature

$A_{i,j,k}$	point d'attachement du segment i avec la plate-forme mobile de l'unité (j, k)
$B_{i,j,k}$	point d'attachement du segment i avec la base de l'unité (j, k)
c_1	coefficient d'accélération, constant
c_2	coefficient d'accélération, constant
C_{obj}	fonction coût
F_i	tension dans le câble i
${}^{RL}_a D_t^{-\alpha}$	intégrale fractionnaire d'ordre α de Riemann-Liouville à droite
${}^{RL}_t D_b^{-\alpha}$	intégrale fractionnaire d'ordre α de Riemann-Liouville à gauche
${}^{RL}_t D_b^\alpha$	dérivée fractionnaire d'ordre α de Riemann-Liouville à droite
${}^C_a D_t^\alpha$	dérivée fractionnaire d'ordre α de Caputo à droite
${}^C_t D_b^\alpha$	dérivée fractionnaire d'ordre α de Caputo à gauche
${}^{GL}_a D_t^\alpha$	dérivée fractionnaire d'ordre α de Grünwald-Letnikov à droite
${}^{GL}_t D_b^\alpha$	dérivée fractionnaire d'ordre α de Grünwald-Letnikov à gauche
$d_{j,k}$	distance radiale
E	module de Young
g	constante de la gravité
I_b	moment d'inertie du disque
I_x	moment d'inertie de l'axe central
k_p	coefficient proportionnel du contrôleur PID
k_i	coefficient intégral du contrôleur PID
k_d	coefficient dérivé du contrôleur PID
$l_{i,k}$	longueur du câble i de la section k
l_k^{min}	longueur minimale du câble i de la section k
l_k^{max}	longueur maximale du câble i de la section k

m	masse du disque
N_p	horizon de prédiction
N_u	horizon de commande
N_1	horizon initial
$\mathbf{n}_{k,l}, \mathbf{b}_{k,l}, \mathbf{t}_{k,l}$	vecteurs unitaires de la matrice R_k^{k-1} définissant le repère \mathfrak{R}_k dans \mathfrak{R}_{k-1}
$O_{j-1,k}$	origine du repère $\mathfrak{R}_{j-1,k}$
$O_{j,k}$	origine du repère $\mathfrak{R}_{j,k}$
$\mathbf{P}_{j,k}^{j-1,k}$	position de $O_{j,k}$ l'origine du repère $\mathfrak{R}_{j,k}$ par rapport au repère $\mathfrak{R}_{j-1,k}$
\mathbf{Q}	forces généralisées
\mathbf{q}	coordonnées généralisées
$R(m)$	partie réelle d'un nombre complexe m
$\mathbf{R}_{j,k}^{j-1,k}$	matrice d'orientation (3×3), définissant le repère $\mathfrak{R}_{j,k}$ dans le repère $\mathfrak{R}_{j-1,k}$
\mathfrak{R}_k	repère relie à l'extrémité de la section k
\mathfrak{R}_0	repère de référence, lie à la plate forme de la première section
$\mathfrak{R}_{j,k}$	repère relie à l'extrémité d'unité j de la section k
$r_{j,k}$	rayon de courbure d'unité j de la section k
t	temps
\dagger	itération
T	énergie cinétique
$\mathbf{T}_{j,k}^{j-1,k}$	matrice de transformation homogène (4×4), définissant le repère $\mathfrak{R}_{j,k}$ dans le repère $\mathfrak{R}_{j-1,k}$
U	énergie potentielle
\mathbf{U}	vecteur de commande
$\mathbf{v}_{k,l}$	vitesse linéaire locale
\mathbf{v}_k	vitesse linéaire globale du disque k
W	ensemble des contraintes sur la sortie et la commande imposées
$x_{j,k}, y_{j,k}, z_{j,k}$	cordonnées cartésiennes par rapport au repère local $\mathfrak{R}_{j-1,k}$
X_{gbest}^\dagger	meilleure position trouvé par l'essaim
X_{pbest}^\dagger	meilleure position par laquelle la particule p est déjà passée
\hat{Y}	sortie du modèle de prédiction
z^{-1}	opérateur de retard

Lettres grecques

θ_k	angle de courbure de la section k
$\theta_{j,k}$	angle de courbure d'unité j de la section k
φ_k	angle du plan contenant l'arc k par rapport à l'axe X
$\kappa_{j,K}$	courbure d'unité j de la section k
$\Theta_{j,k}$	angle de tangage, correspond à la rotation autour de l'axe $y_{j,k}$
$\Psi_{j,k}$	angle de roulis, correspond à la rotation autour de l'axe $x_{j,k}$
$\Gamma(\cdot)$	fonction Gamma
λ	ordre d'intégration fractionnaire
μ	ordre de dérivée fractionnaire
$\omega_{k,l}$	vitesse angulaire locale du disque
ω_k	vitesse angulaire globale du disque k

Acronymes

CBHA	Combact Bionic Handling Arm
CDCR	Cable-Driven Continuum Robot
CC	Courbure Constante
FOPID	Fractional Order Proportional –Integral –Derivative
<i>IAE</i>	Intégrale de la valeur absolue de l’erreur (Integral of Absolute Error)
<i>ISE</i>	Intégrale du carré de l’erreur (Integral of Squared Error)
<i>ITAE</i>	Intégrale de la valeur absolue de l’erreur pondérée par le temps (Integral of Time multiplied by Absolute Error)
<i>ITSE</i>	Intégrale du carré de l’erreur pondérée par le temps (Integral of Time Squared Error)
MAE	Erreur absolue moyenne (Mean of Absolute Error)
MGD	Modèle Géométrique Direct
MGI	Modèle Géométrique Inverse
MLP	Perceptron multicouches (Multi-Layer-Perceptron)
MSE	Erreur quadratique moyenne (Mean Squared Error)
NPMC	Non-linear Model Predictive Control
OPID	Contrôleur PID optimisé par PSO
PID	Proportional –Integral– Derivative
PMC	Commande prédictive à base de modèle (Model Predictive Control)
RBF	Radial Basis Function
RTC	Robots à Tube Concentrique
SQP	Programme Quadratique Séquentiel (Sequential Quadratic Program)
UPS	Universal joint - Prismatic joint - Spheric joint
UP	Universal joint - Prismatic joint

Introduction générale

Sommaire

A.	Motivation et contexte de la thèse	2
B.	Objectifs de la thèse	2
C.	Contributions	3
D.	Résultats obtenus	3
E.	Organisation du manuscrit	4

De nos jours, les robots manipulateurs à liaisons rigides sont largement utilisés dans divers secteurs industriels pour assister les êtres humains dans des tâches dangereuses, monotones, fatigantes ainsi que celles nécessitant une grande précision. Cependant, cette catégorie de robots souffrent de certaines limitations telles que : une interaction physique dangereuse homme-robot, une incapacité à s'adapter à leur environnement en raison de leur structure rigide ainsi que de leur manœuvrabilité réduite. Dans leur volonté d'améliorer les performances de cette catégorie, les chercheurs ont montré un vif intérêt pour l'étude et le développement d'architectures mécaniques robotisées directement inspirées des organismes vivants. Cette catégorie de robots est appelée robots bio-inspirés ou plus généralement robots continuum.

Un robot continuum ne possède ni liaisons discrètes ni corps rigides dans son architecture, à l'exception de quelques types de robots actionnés par câbles/tendons. Son corps est souple et flexible semblable à certains appendices et tentacules animales tels que : les trompes d'éléphants, les céphalopodes et les serpents, ainsi que certaines plantes assurant le processus de croissance comme la vigne-plante. Cette nouvelle catégorie de robots fait actuellement l'objet de nombreux travaux de recherche compte tenu de leur hyper-redondance, la dextérité et la flexibilité structurelle plus élevée. Ces caractéristiques rendent les robots continuum plus adaptés à une variété d'applications notamment : les applications médicales et chirurgicales, l'inspection et la manipulation, etc.

La modélisation des robots continuum, qu'ils soient souple ou rigide, pose plusieurs défis par rapport aux robots manipulateurs à liaisons rigides. Ces défis sont principalement dus à la redondance cinématique, la non-linéarité et à la grande flexibilité de leur structure du robot. Par conséquent, la modélisation de ce type de robots nécessite d'avoir un modèle continu de l'axe central du robot. Compte tenu de cette complexité, la plupart des modèles mathématiques développés à ce jour sont basés sur des hypothèses et des approximations. En dépit de ces simplifications, le calcul du modèle géométrique inverse des robots continuum, notamment la résolution de sa redondance reste toujours un défi. D'autre part, en raison de la non-linéarité des équations mathématiques des modèles qui régissent le comportement cinématique et dynamique de ce type de robots, rend l'implémentation de ces modèles problématique, notamment en vue d'une commande réaliste.

A. Motivation et contexte de la thèse

Les robots manipulateurs à liaisons rigides sont utilisés dans une grande variété d'applications, mais les tâches qu'ils peuvent gérer sont limitées et particulièrement peu sûres en termes d'interaction homme-robot. En revanche, les robots continuum peuvent être utilisés pour de nombreuses applications et dans des environnements complexes et encombrés en raison de leur grande flexibilité et dextérité. Cependant, la modélisation et la commande de ces types de robots en sont encore à leurs balbutiements, ce qui empêche leur déploiement et la pleine exploitation de leurs potentiels. Dans ce contexte, le travail envisagé dans cette thèse consiste à concevoir et à développer des modèles mathématiques en vue de contribuer à la résolution des problèmes liés à la modélisation géométrique inverse et sa redondance des robots continuum, ainsi que l'implémentation de quelques schémas de commande en vue d'une commande réaliste de ces types de robots.

B. Objectif de la thèse

L'objectif principal de cette étude est de contribuer à la résolution du problème du modèle géométrique inverse et sa redondance des robots continuum, en particulier le robot continuum actionné par câbles, en Anglais "Cable-Driven Continuum Robot (CDCR)". D'autre part, en raison de la complexité des modèles mathématiques qui régissent le mouvement et le comportement dynamique des robots continuum, le deuxième objectif de cette thèse est de mettre en œuvre certains schémas de commande. A cette fin, le travail est divisé en deux étapes principales. Dans la première partie consacrée à la modélisation, un algorithme fiable est proposé pour résoudre le problème du modèle géométrique inverse et sa redondance en se

basant sur le modèle géométrique direct développé sous l'hypothèse à courbure constante. L'algorithme proposé se compose de deux sous-algorithmes. Le premier sous-algorithme est alloué pour déterminer toutes les solutions possibles du modèle géométrique inverse, tandis que le second est consacré à la résolution du problème de la redondance. La dernière étape consiste à concevoir et à mettre en œuvre quelques schémas de commande, à savoir le contrôleur : Proportionnel-Intégral Dérivé (PID), PID Optimisé (OPID), PID d'Ordre Fractionnaire (FOPID) et la commande prédictive non linéaires (NMPC), afin d'assurer le suivi de trajectoire via la technique point-à-point, tant pour le modèle cinématique que dynamique du robot en question.

C. Contribution

Les contributions majeures apportées dans cette thèse peuvent se résumer dans les points suivants :

- ✓ Développement d'un algorithme fiable capable de résoudre le problème du modèle géométrique inverse et sa redondance des robots continuum. L'algorithme proposé se compose de deux sous-algorithmes.
 - Le premier sous-algorithme introduit une nouvelle solution qui permet de réduire le nombre infini de solutions du modèle géométrique inverse à un nombre fini en prenant en compte les ensembles de poses accessibles des premières $(n - 1)$ sections flexibles, qui sont sélectionnées et créées une-par-une de la première à la $(n - 1)$ ème section flexible, dans laquelle la méthode d'optimisation par essaim des particules est adoptée pour sélectionner la première solution aléatoire du problème inverse.
 - Le second sous-algorithme est consacré à la résolution de la redondance dans lequel un critère spécifique a été accordé pour sélectionner la solution appropriée parmi les variétés des solutions existantes.
- ✓ Proposition et implémentation de quelques schémas de commande, à savoir : Proportionnel-Intégral-Dérivé (PID), PID optimisé (OPID), PID d'Ordre Fractionnaire (FOPID) et la commande prédictive non linéaires (NMPC), afin d'assurer le suivi de trajectoire via la technique point-à-point, tant pour le modèle cinématique que dynamique du robot en question.

D. Résultats obtenus

Les résultats obtenus ont fait l'objet des publications et communications suivantes :

✓ **Publication internationale**

- Merrad, A., Amouri, A., Cherfia, A. et al. "A Reliable Algorithm for Obtaining All-Inclusive Inverse Kinematics' Solutions and Redundancy Resolution of Continuum Robots". *Arabian Journal for Science and Engineering*, Vol. 48, no. 3, pp. 3351–3366 (2023). <https://doi.org/10.1007/s13369-022-07065-0>

✓ **Communications internationales**

- Merrad, A., Amouri, A., Cherfia. "An optimization-based approach for solving inverse kinematics and redundancy resolution of multi-section continuum robots". *International Conference on Mechanical Sciences*, November 15-16, 2021, Larbi Ben M'Hidi University, Oum El Bouaghi, Algeria.
- Merrad, A., Amouri, A. "Fractional-Order PID Control of a Continuum Robot". *The 4th International Conference on Electromechanical Engineering (ICEE'2022)*, November 22-23, 2022. Skikda University, Algeria.
- Amouri, A., Bousbia, L., Merrad, A. A Nonlinear Model Predictive Control Based Particle Swarm Optimization for the Trajectory Tracking of a Cable-Driven Continuum Robot. *The 4th International Conference on Electromechanical Engineering (ICEE'2022)*, November 22-23, 2022. Skikda University, Algeria.

E. Organisation de la thèse

Cette thèse est organisée en quatre chapitres comme suit :

- Le premier chapitre fournit une vue d'ensemble des robots continuum, en proposant des classifications pour ces robots ainsi que les techniques de modélisation couramment utilisées pour ces systèmes bio-inspirés, en plus des schémas de commande proposés dans la littérature. Ces sections du chapitre nous aident à situer notre travail par rapport à la littérature existante sur la modélisation et la commande des robots continuum. De plus, le chapitre examine les différents types de robots continuum, discute de leurs avantages et inconvénients, ainsi que de leurs applications et domaines d'utilisation.
- Le chapitre 2 est consacré à la modélisation géométrique d'un robot continuum actionné par câbles, nommé en Anglais : Cable-Driven Continuum Robot (CDCR). Initialement, une description géométrique du robot en question est présentée. La première partie est consacrée au développement mathématique du modèle géométrique direct du robot. Le modèle a été développé sous l'hypothèse de courbure constante tout en gardant l'axe

central du robot inextensible. La deuxième partie est dédiée à la résolution du modèle géométrique inverse du robot en question. Pour ce faire, un algorithme fiable est développé. Cet algorithme se compose de deux sous-algorithmes. Le premier sous-algorithme est alloué pour déterminer toutes les solutions possibles du modèle géométrique inverse, dans lequel la première solution aléatoire est calculée à l'aide d'une méthode d'optimisation, à savoir : optimisation par essaim de particules. La dernière partie de chapitre présente les résultats de simulation.

- Le chapitre 3 se concentre sur la résolution du problème de la redondance du modèle géométrique inverse en utilisant le second sous-algorithme. Celui-ci met en œuvre un critère spécifique pour sélectionner la solution la plus adaptée parmi les différentes solutions existantes. Les résultats de simulations sont ensuite présentés à la fin de ce chapitre.
- Le chapitre 4 est consacré à la commande des robots continuum. Dans un premier temps, les modèles cinématiques et dynamiques des robots continuum étudiés ont été présenté. Puis, les principes de base des quatre schémas de commande nommément PID, OPID, FOPID et NMPC sont donnés. Ensuite, Les quatre schémas proposés ont été implémentés dans l'environnement Matlab et les résultats de simulation obtenus ont été comparés.

La thèse se termine par une conclusion générale qui regroupe de façon succincte les résultats obtenus ainsi que les perspectives de recherche possible.

Chapitre 1

Généralités sur les robots continuum

Sommaire

1.1.	Introduction	7
1.2.	Les robots continuum	7
1.3.	Classification des robots continuum	11
1.3.1.	Classification selon la structure du robot	11
1.3.2.	Classification selon l'endroit des actionneurs	12
1.4.	Applications et défis des robots continuum	13
1.5.	Avantages et inconvénient des robots continuum	14
1.6.	Approches et méthodes de modélisation	15
1.6.1.	Modèles géométriques	15
1.6.2.	Modèles dynamiques	17
1.7.	Stratégies de commande	18
1.8.	Conclusion	18

1.1. Introduction

Dans la première partie de ce chapitre, nous montrons d'abord l'importance d'utiliser des robots continuum par rapport à leurs homologues à liaisons rigides, puis nous présentons une recherche bibliographique sur l'évolution de ces robots, leurs classifications et les domaines d'application. Dans la deuxième partie, nous présentons l'état de l'art sur les méthodes et les approches de modélisation ainsi que les schémas de commande utilisés pour ces types de robots afin de positionner nos travaux dans le cadre de la littérature.

1.2. Les robots continuum

Les manipulateurs à liaisons rigides ont des structures mécaniques constituées d'une série de corps rigides reliés entre eux par des articulations prismatiques et/ou rotoïdes [1]. Typiquement, un robot manipulateur est monté au sol pour des raisons de sécurité et ses liaisons rigides discrètes sont actionnées pour effectuer un mouvement spatial de leur effecteur terminal. Un logiciel de contrôle sophistiqué fait fonctionner des robots individuellement ou coordonne plusieurs robots afin de maximiser leur valeur dans un processus industriel donné. Bien que ces robots aient très bien réussi à accomplir leurs tâches dans de nombreux secteurs de l'industrie (automobile, agroalimentaire, pharmaceutique, etc.) [2-4], et aient réussi à relever un grand nombre de défis (qualité, précision et rapidité, meilleure ergonomie du poste de travail, réduction des coûts de main-d'œuvre, amélioration de la production, etc.) [5-6], ils restent toujours souffrent de nombreuses lacunes et limitations, notamment : l'interaction physique non sécurisée homme-robot, l'incapacité de s'adapter à leur environnement en raison de sa structure rigide et leur manœuvrabilité réduite [7-8]. Dans ce contexte, les robots continuum émergent comme un nouveau concept, au moins dans la recherche, avec le potentiel d'être utilisé dans un large éventail d'applications industrielles [9].

Les robots continuum sont des structures électromécaniques hyper-flexibles à degrés de liberté infinis qui leur confèrent la capacité de manœuvrer des trajectoires curvilignes complexes [10]. Un avantage clé des robots continuum par rapport à ceux à liaisons rigides est que, en raison de leur poids considérablement faible pour la même force de sortie maximale, ils peuvent être montés au plafond plutôt qu'au sol. Cet avantage augmente considérablement leur sécurité lorsqu'ils sont utilisés dans des opérations conjointes avec des humains sur le sol de l'usine [11]. D'un autre côté, les robots continuum sont intrinsèquement

non linéaires et donc plus difficiles à contrôler que leurs homologues à liaisons rigides, présentant ainsi un obstacle à l'adoption dans l'industrie [12].

Les robots continuum ont une flexibilité accrue, et donc une grande dextérité, par rapport à leurs homologues à liaisons rigides, Ce qui montre l'importance des robots continuum. La Figure 1.1 illustre la différence essentielle entre une structure discrète, serpentine et à liaison continue [13-14]. Il ressort clairement de cette figure que les structures continuum ont plus de degrés de liberté pour se déplacer et, par conséquent, elles sont capables de se déplacer plus précisément le long de la forme d'un objet, Figure 1.2. De plus, leurs extrémités peuvent se positionner dans beaucoup plus de directions par rapport aux structures à liaisons rigides.

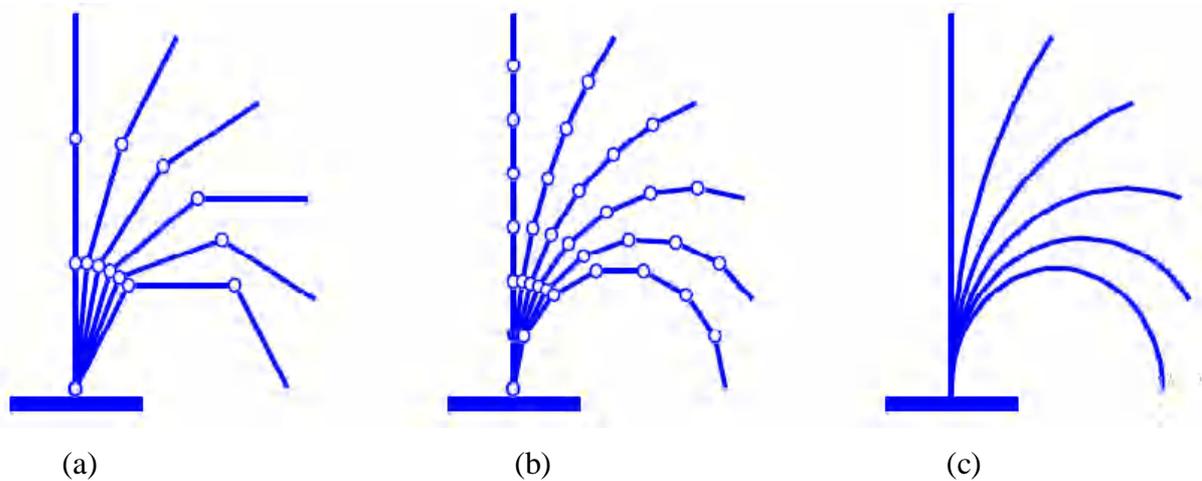


Figure 1.1 : Différenciation : liée et continue. (a) structure à liaisons discrète rigide, (b) structure serpentine, et (c) structure continue [13].

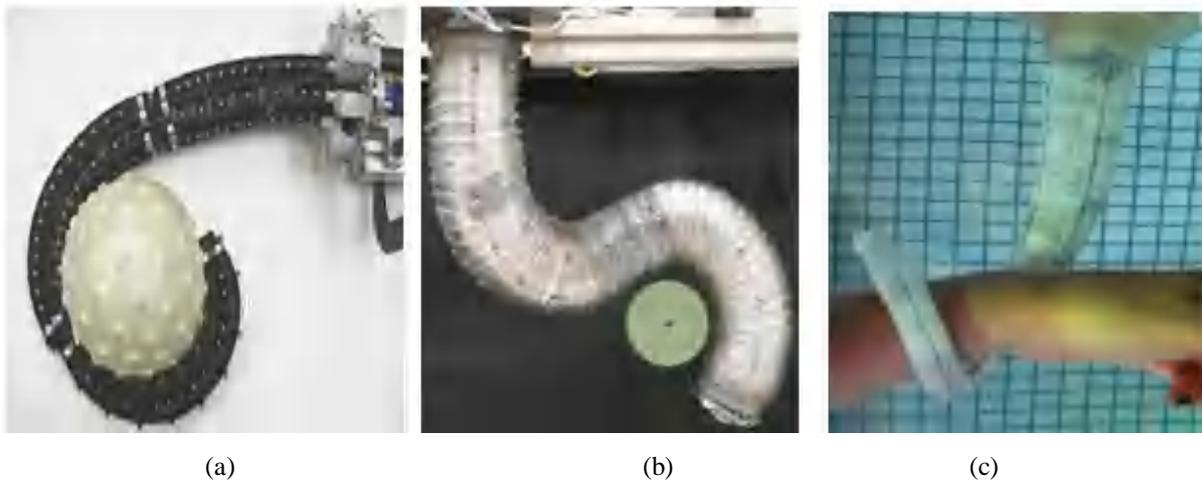


Figure 1.2 : Exemples de saisie d'objets : (a) Robot OctArm IV [15], (b) Robot Air-Octor [16], Octopus-Arm [17].

Les premiers robots continuum ont été développés dans les années 1960 (Figure 1.3) et ont pris de l'importance à la fin des années 1990. Le Tableau 1.1, passe en revue l'évolution des robots continuum conçus et leurs analogues dans les organismes vivant.

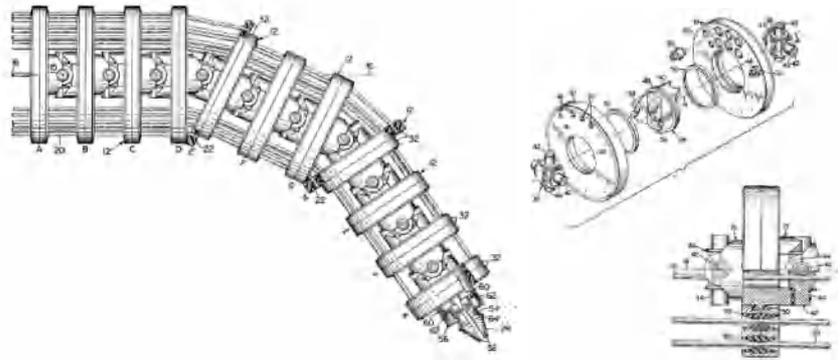


Figure 1.3 : Tensor Arm manipulator [18].

Tableau.1.1 : Comparaison de quelques robots continuum avec leur bio-inspiration.

Bio-inspiration	Modèle	Exemple (robot continuum)	Réf.
Bras et doigt humains			[19-20]
Colonne vertébrale d'un mammifère			[21]
Langue et queue d'un reptile			[22-23]
Trompe d'éléphant			[24]

Bio-inspiration	Modèle	Exemple (robot continuum)	Réf.
Serpent			[25]
Bras de poulpe			[26-27]
Os de poisson			[28]
Tentacule			[29]
Vigne et plante biologique			[30-33]

1.3. Classification des robots continuum

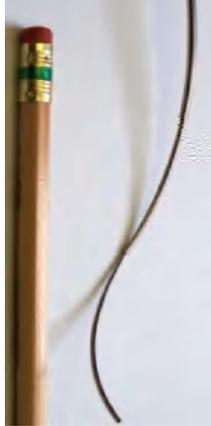
Les robots Continuum peuvent être classés selon deux critères principaux : la structure et la et l'emplacement de leurs actionneurs.

1.3.1. Classification selon la structure du robot

La caractéristique principale de la conception des robots continuum est la présence d'une structure centrale à courbure continue, appelée dorsale. Selon les principes de conception approuvés pour les robots continuum, on distingue :

- Epine dorsale unique : ce type de robots a une épine dorsale élastique centrale à travers laquelle les éléments d'actionnement/transmission peuvent fonctionner.
- Multi-épine dorsale : la structure de ces robots continuum comporte deux ou plusieurs éléments élastiques (tiges ou tubes) parallèles et contraints en même temps les uns aux autres.

Tableau 1.2 : Quelques types de robots continuum classés selon leur structure.

Epine dorsale unique	Réf.	Multi-épine dorsale	Réf.	Tube concentrique	Réf.
	[27]		[35]		[36]
	[34]		[34]		[34]

- Tube concentrique : l'épine dorsale de ce type de robots continuum est constituée de tubes concentriques qui sont libres de tourner et de se translater entre eux, en fonction de l'actionnement qui se produit à la base du robot.

Tableau 1.2 présente quelques types de robots continuum classés selon le type de leur structure.

1.3.2. Classification selon l'endroit des actionneurs

Selon l'endroit où l'actionnement se produit, on distingue trois types de stratégie d'actionnement des robots continuum : actionnement extrinsèque, intrinsèque et hybride.

- Actionnement extrinsèque : l'actionnement se fait à l'extérieur de la structure flexible ou souple du robot et les forces d'actionnement sont transmises par transmission mécanique soit par des câbles ou des tendons (Figure 1.4).

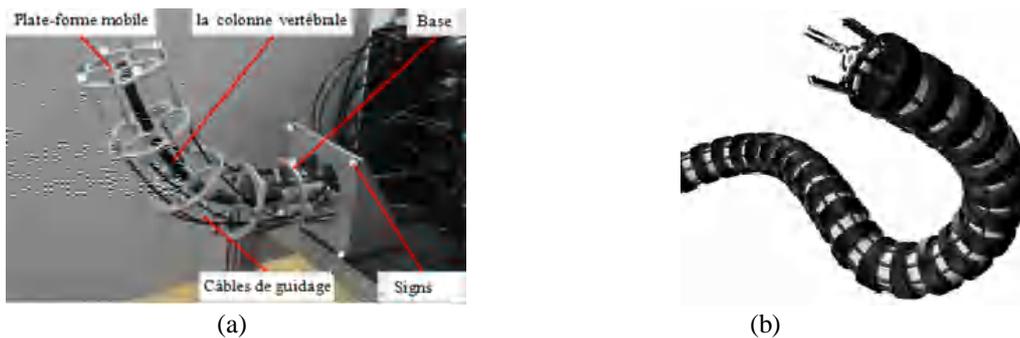


Figure 1.4 : Exemples d'actionnement extrinsèque : (a) robot actionné par câbles [37], (b) robot actionné par tendons [38].

- Actionnement intrinsèque : le mécanisme d'actionnement est intégré dans la structure du robot. Cette stratégie inclue les chambres pneumatiques, hydrauliques et l'effet de mémoire de forme (Figure 1.5).

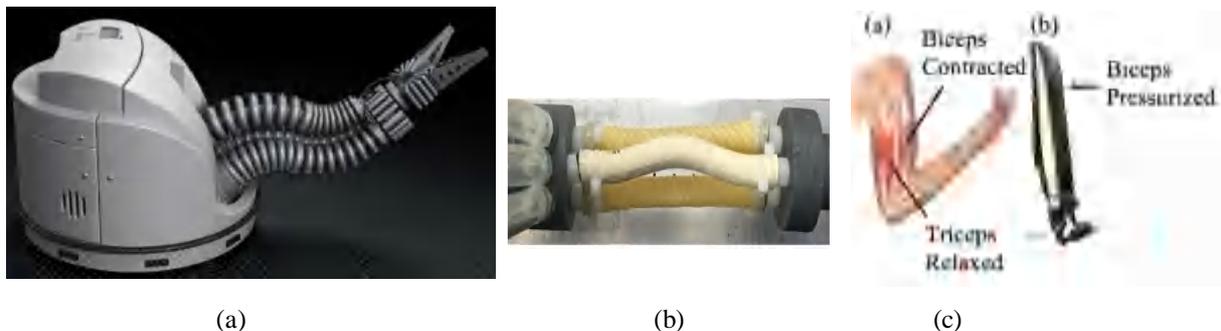


Figure 1.5 : Exemples d'actionnement intrinsèque : (a) actionnement pneumatiques [39], (b) actionnement musculaire pneumatique artificiel (PMA) [40], et (c) PMA tressé [41].

- Actionnement hybride : Le robot a deux mécanismes, un intrinsèque et un extrinsèque. Généralement, la cavité centrale du robot est considérée comme l'actionneur intrinsèque et l'actionneur extrinsèque est soit des câbles ou des vérins hydrauliques (Figure 1.6).

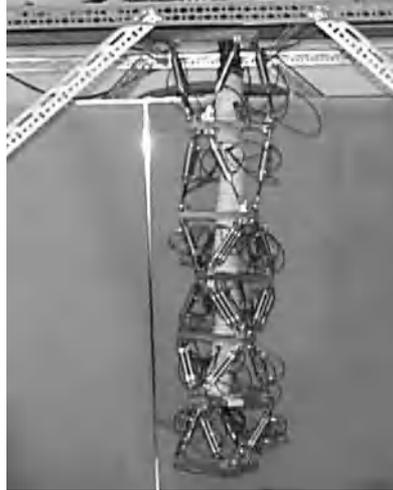


Figure 1.6 : Exemple d'actionnement extrinsèque et intrinsèque [42].

1.4. Applications et défis des robots continuum

La polyvalence des robots continuum a renforcé leur attractivité pour de nombreuses applications différentes, telles que la navigation, l'assemblage, la fabrication et même dans le domaine de la médecine, à savoir en chirurgie. Le tableau 1.3 résume quelques fonctions d'application des robots continuum.

Tableau 1.3 : Quelques fonctions d'applications des robots continuum.

Structure du robot	Description	Application	Réf.
Un segment flexible (2 à 3-DDL)	Le corps du robot se compose d'une seule structure en forme de colonne vertébrale à mouvement limité.	Nettoyage	[43-44]
Multi-segment flexible (n -DDL)	Le corps du robot se compose de plusieurs segments, chacun indépendamment commandé des autres.	Inspection et exploration	[45-49]
Une section flexible rigide (2 à 3-DDL)	Le corps du robot est composé de plusieurs disques de rayons identiques ou variables, disposés à distance égale les uns des autres, et l'épine dorsale du robot est un segment unique.	Médecine et chirurgie	[50-52]

Structure du robot	Description	Application	Réf.
Multi-section flexible rigide (n -DDL)	le corps du robot est composé de plusieurs disques de rayons identiques ou variables disposés à distance égale les uns des autres, la structure dorsale est composée de plusieurs segments.	Mobilité	[53-56]
Structure continue (n -DDL)	Le corps du robot est constitué de plusieurs fibres tressées ensemble. Les extrémités des fibres sont connectées à des câbles qui peuvent être enroulés/déroulés individuellement à partir d'une poulie.	Manipulation	[57-59]

1.5. Avantages et inconvénients des robots continuum

La conception particulière des robots continuum offre plusieurs avantages par rapport aux robots à liaisons rigides. En revanche, leur conception particulière introduit également de nombreux défis. Le Tableau 1.4 résume quelques avantages et inconvénients des robots continuum.

Tableau 1.4 : Quelques avantages et inconvénients des robots continuum.

Avantages	Inconvénients
Rapport diamètre/longueur relativement faible : ce qui permet aux robots continuum de naviguer et opérer dans des espaces confinés.	Risque de lésions nerveuses, notamment pour les robots continuum utilisés en chirurgie.
Flexibilité : s'adapter facilement à se déformer en fonction de l'environnement.	Modèles mathématiques complexes et hautement non linéaires.
La douceur "the softness" : le matériau fréquemment utilisé garantit une plus grande sécurité dans les interactions homme-robot.	la compréhension du comportement et des propriétés des robots continuum fait toujours l'objet des recherches.
Dextérité : fonctionnement facile dans des environnements complexe et encombré.	Implémentation difficile de schémas de commande

Le plus important que l'essor des robots continuum a également ouvert la voie au développement de manipulateurs continuum souples.

1.6. Approches et méthodes de modélisation

D'un point de vue modélisation, il existe trois modèles mathématiques pour décrire le comportement d'un tel robot. Ces modèles sont résumés comme montre la Figure 1.7. Dans cette partie, on présente quelques méthodes et approches permettant d'établir ces modèles.

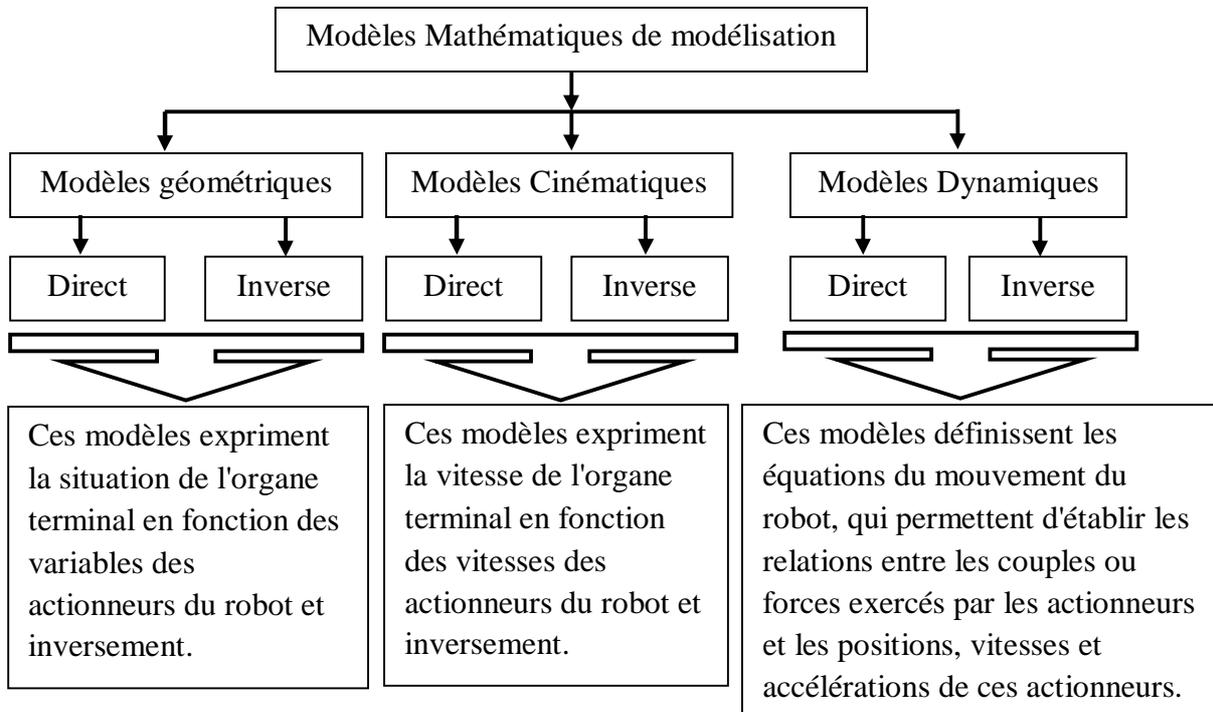


Figure 1.7 : Modèles mathématique de modélisation des robots.

1.6.1. Modèles géométriques

La résolution du problème des modèles géométriques directs des robots continuum est relativement facile à résoudre. En conséquence, de nombreuses contributions ont été apportées en utilisant différentes méthodes et théories, dont la plupart sont basées sur certaines hypothèses et simplifications, à savoir l'approche de l'hypothèse de courbure constante [60]. Cette approche est basée sur la formulation mathématique d'un arc de cercle. Le Figure 1.8 résume quelques techniques couramment utilisées pour le calcul des modèles géométriques directs des robots continuum.

En revanche, la résolution du problème du Modèle Géométrique Inverse (MGI) des robots continuum est encore assez difficile en raison des équations non linéaires et fortement couplées des modèles géométriques directs. En fait, il est difficile de résoudre ce problème de manière purement analytique sauf dans des cas particuliers où le robot répond à des conditions particulières, comme dans [75-78].

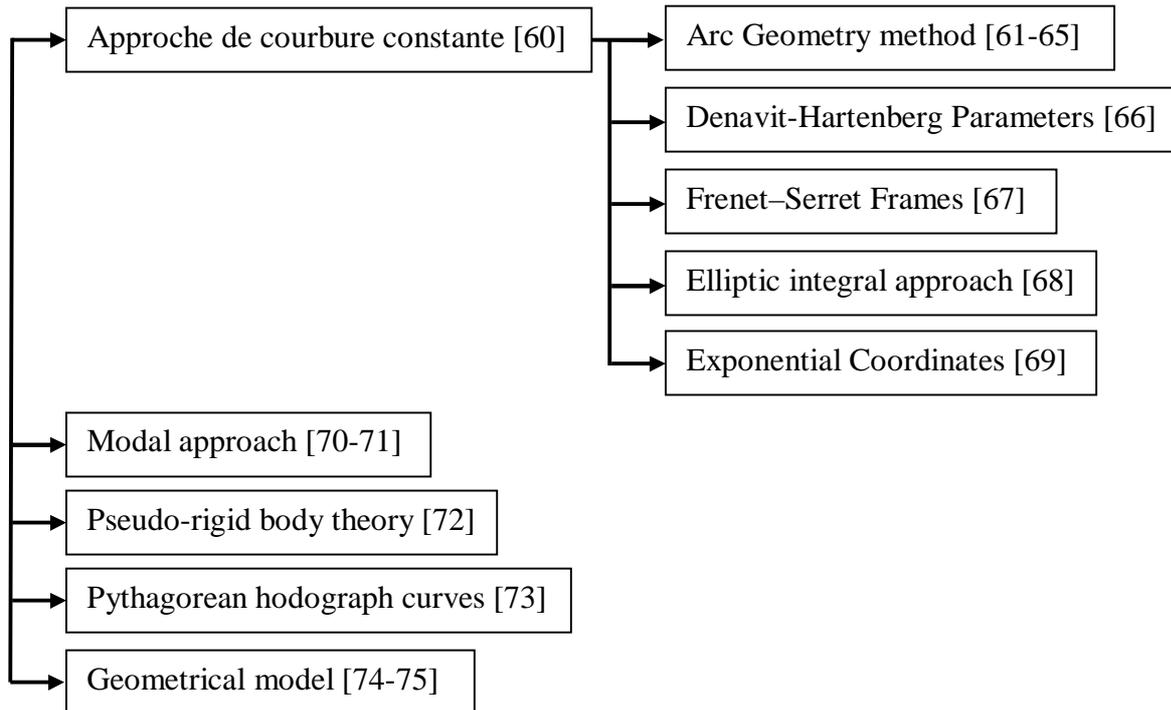


Figure 1.8 : Méthodes et approches de résolution des modèles géométriques directs des robots continuum.

Dans la référence [73], le problème du MGI du robot CBHA a été résolu à l'aide de courbes d'hodographe de Pythagore, dans lesquelles la forme du robot est reconstruite en courbes afin de dériver le modèle. Cependant, cette approche ne peut pas traiter le problème de la redondance ; ainsi elle ne peut donc pas être utilisée pour trouver des solutions générales au problème.

Autres chercheurs ont utilisé la courbe d'ajustement pour résoudre le problème du MGI des robots continuum [79-82]. Par exemple, dans [79], les auteurs ont utilisé une courbe particulière pour modéliser un robot hautement redondant. Cependant, l'approche correspond à peu de robots continuum, ce qui limite son applicabilité. Dans la même veine, les auteurs de [80] ont étendu l'approche [79] en introduisant une courbe plus ajustée qui se rapproche de la forme du robot hyper-redondant réel. Aussi, dans la référence [28], les auteurs ont utilisés des fonctions polynomiales pour résoudre le MGI d'un robot continuum actionné par câbles à deux modules, chacun à une déformation planaire.

Alternativement, certains chercheurs ont eu recours à des outils d'apprentissage pour résoudre le MGI des robots continuum. Par exemple, dans la référence [81], une approche de réseaux de neurones via une architecture d'apprentissage supervisé distal a été utilisée pour résoudre le problème inverse du manipulateur CBHA. Dans le même contexte, les auteurs de

[82] ont utilisés une approche hybride dans lequel le manipulateur CBHA était considéré comme une série de vertèbres. Cependant, ces approches ne peuvent fournir qu'une seule solution au MGI qui rend ces approches inutilisables dans des cas généraux.

Autres chercheurs ont utilisé des méthodes numériques et d'optimisation pour résoudre le problème du MGI des robots continuum. Dans [13], [28], [83], le problème du MGI pour un robot continuum actionné par câbles, ou en Anglais Cable-Driven Continuum Robot (CDCR), planaire et spatial a été formulé comme un problème d'optimisation, puis le problème est résolu par la méthode PSO. Dans [84], les auteurs ont utilisés une approche basée sur le programme quadratique séquentiel (SQP) pour approximer le MGI du manipulateur CBHA. Cependant, ces approches, ne peuvent fournir qu'une seule solution optimale ou non optimale au problème. De même, des méthodes numériques basées sur la matrice jacobienne ont été mises en œuvre pour résoudre le problème du MGI des robots continuum [71], [85-86], dans lequel les équations du MGD sont différenciées pour trouver une transformation linéaire entre les vitesses des actionneurs et les vitesses de la pose finale du robot.

Sur la base des méthodes et des approches précédemment proposées pour résoudre le problème du MGI des robots continuum, nous avons remarqué que la majorité des chercheurs n'ont pas abordé à fond la résolution de la redondance, qui est indéniablement importante.

1.6.2. Modèles dynamiques

En réalité, l'utilisation des modèles géométriques pour décrire le comportement ou la pose d'un robot continuum est insuffisante du fait que les chargements internes et externes agissant sur le robot jouent un rôle important dans la configuration du robot. Par exemple, la gravité du robot peut provoquer des déviations importantes dans la courbure du robot même dans l'espace libre et lorsqu'aucune charge n'est portée. Les modèles dynamiques qui visent à établir les relations entre les couples ou forces exercés par les actionneurs et les positions, vitesses et accélérations de ces actionneurs, ont fait l'objet d'intenses recherches dans ces dernières années. Ainsi, plusieurs méthodes, principe et formalismes ont été utilisés, à savoir : Le formalisme d'Euler-Lagrange, le principe de la puissance virtuelle (la méthode de Kane), le principe de Hamilton et la théorie de la tige de Cosserat. Par exemple, le formalisme d'Euler-Lagrange a été employé pour modéliser plusieurs types de robots continuum tel que : robot continuum à multiple-épines dorsales [87], BHA [88] et CDCR [13, 89-91]. Le principe de la puissance virtuelle a été utilisé pour modéliser la dynamique d'une seule section pour un manipulateur continuum entraîné tenant compte des effets inertiels, de frottement et

gravitationnels [92]. Les mêmes auteurs ont étendu ce modèle au robot continuum multi-sections [93]. Le principe de Hamilton a été utilisé pour dériver la dynamique d'un robot continuum planaire dans les grandes déformations [94]. Autres chercheurs ont utilisés la théorie de Cosserat pour dériver les modèles dynamiques des robots continuum rigides et souples [95-96].

1.7. Stratégies de commande

De nos jours, la commande des robots continuum reste une tâche laborieuse et l'un des axes de recherche majeurs pour la communauté robotique. Malgré cela, de grands efforts ont été déployés pour appliquer des techniques de commande classiques et avancées afin de résoudre le problème de contrôle. En concentrant la commande des CDCRs, seuls quelques travaux ont été proposés, à savoir [97-98]. Dans ces références, les auteurs ont utilisés la commande prédictive non linéaire (NMPC) pour résoudre les problèmes de suivi de trajectoire et d'évitement d'obstacles utilisant des modèles cinématiques et dynamiques. Dans la référence [99], les auteurs ont appliqué le mode glissant non linéaire pour contrôler l'extrémité du CDCR dans le cas planaire. D'autres chercheurs ont utilisés des contrôleurs PID classiques, PID optimisé par la méthode PSO (OPID) et PID d'ordre fractionnaire (FOPID) pour assurer le suivi de trajectoire [13, 89, 91, 100].

D'une manière générale, plusieurs travaux ont été menés. Par exemple, dans [101], la logique floue basée sur le modèle cinématique est proposé pour contrôler l'effecteur terminal du robot continuum actionné par tendons (TDCR). Aussi, dans [102], un contrôleur de rétroaction statique basé sur la logique floue est développé pour une seule section flexible d'un TDCR. De plus, une méthodologie de logique floue a été proposée dans [102-104] pour concevoir un contrôleur non linéaire pour réguler l'effecteur terminal des manipulateurs continus. D'autres chercheurs ont utilisé des schémas de contrôle adaptatifs [81,105]. Le schéma de réseaux de neurones adaptatifs [81] et le contrôleur de régresseur de vecteur de support adaptatif [105] ont été réalisés sur un manipulateur continu, à savoir CBHA.

1.8. Conclusion

Dans ce chapitre, un aperçu détaillé de la littérature sur la conception, la modélisation et la commande des robots continuum a été donné. Dans la première partie, les principaux types de robots continuum ont été présentés, ainsi que leurs classifications selon le type de la structure et l'emplacement de leurs actionneurs. En plus, les avantages, les inconvénients et le

domaine d'application de ces robots sont donnés. Dans la deuxième partie, les méthodes et les approches de modélisation géométrique et dynamique de ce type de robots ont été présentées, en plus des stratégies de commande couramment utilisées. Cet état de l'art nous a permis de découvrir les problématiques et difficultés de développement de ces robots continuum, notamment en termes de modélisation et de commande, ainsi que d'identifier des lieux potentiels d'apport d'intervention pour contribuer à résoudre certains de ces problèmes.

Chapitre 2

Modélisation Géométrique

Sommaire

2.1. Introduction	22
2.2. Description géométrique de la structure du robot	22
2.3. Hypothèses de modélisation.....	24
2.4. Modélisation géométrique directe.....	25
2.4.1. Modèle géométrique direct d'une unité (j, k)	26
2.4.2. Modèle géométrique direct d'une section flexible k	30
2.4.3. Modèle géométrique direct d'un robot continuum multi-sections	30
2.5. Modélisation géométrique inverse.....	30
2.5.1. Modèle géométrique inverse d'une unité (j, k)	31
2.5.2. Modèle géométrique inverse d'une section k	32
2.5.3. Modèle géométrique inverse d'un robot continuum multi-sections	32
2.5.3.1. L'algorithme PSO	34
2.5.3.2. La fonction de coût	36
2.5.3.3. Sélection des paramètres de l'algorithme PSO	36
2.5.3.4. Condition aux limites	36
2.5.3.5. Limitations physiques	37
2.5.4. Application	37
2.5.4.1. Application 1	37
2.5.4.2. Application 2	39
2.6. Simulations numériques	41
2.6.1. Détermination de l'espace de travail	42

2.6.1.1.	Détermination de l'espace de travail d'un robot à une section flexible	42
2.6.1.2.	Détermination de l'espace de travail d'un robot multi-sections ..	44
2.6.2.	Simulation des modèles géométrique inverse utilisant le sous-algorithme 1	44
2.6.3.	Contributions récentes sur la modélisation géométrique inverse des robots continuum	49
2.7.	Conclusion.....	50

2.1. Introduction

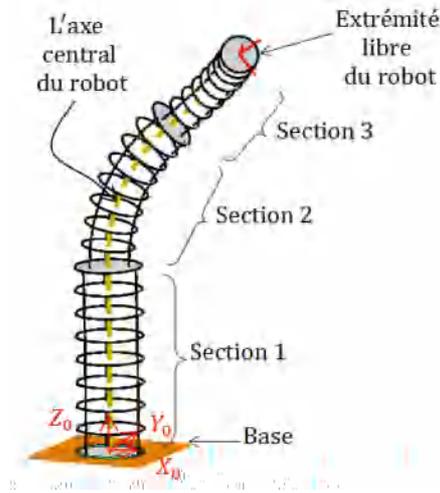
La modélisation géométrique est une étape essentielle et incontournable dans le développement des modèles cinématiques et dynamiques de n'importe quel système robotique. Physiquement, la modélisation géométrique consiste à décrire le mouvement par rapport à un référentiel cartésien fixe en ignorant les forces et les moments qui provoquent le déplacement de la structure. La formulation des relations géométriques permet l'étude de deux problèmes clés de la robotique, à savoir : le problème de la modélisation géométrique directe et le problème de la modélisation géométrique inverse. Le premier concerne la détermination d'une méthode systématique et générale pour décrire le mouvement de l'extrémité libre du robot, c'est-à-dire l'organe terminal du robot, en fonction des coordonnées généralisées de fonctionnement. Le dernier concerne le problème inverse ; sa solution est d'une importance fondamentale pour transformer le mouvement souhaité, naturellement prescrit à l'organe terminal du robot dans l'espace de travail, en le mouvement des variables de fonctionnement correspondant.

Contrairement aux robots à liaisons rigides, les robots continuum sont difficiles à modéliser en raison du grand nombre, ou plutôt infini, de degrés de liberté requis les modèles géométriques pour décrire le comportement et la forme continue de la structure du robot. Cependant, sur la base de certaines hypothèses et approximations, les modèles géométriques des robots continuum peuvent être établies. Ainsi, l'objectif principal de ce chapitre est de développer le modèle géométrique direct d'un robot continuum, nommé "robot continuum actionné par câbles", ou "en Anglais Cable-Driven Continuum Robot (CDCR)". Dans le même contexte, la solution du modèle géométrique inverse sera le deuxième objectif majeur poursuivi au cœur de ce chapitre.

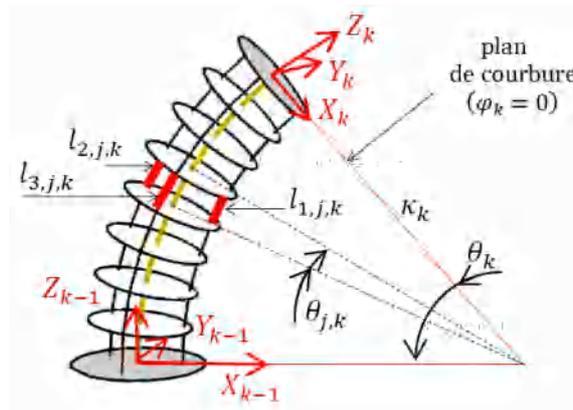
2.2. Description géométrique de la structure du robot

Aux fins de l'analyse géométrique, on peut considérer un tel robot continuum comme un ensemble (n) de sections flexibles connectées en série. La Figure 2.1(a) présente un robot continuum actionné par câbles à trois sections flexibles qui sera le type de robots continuum considéré dans cette étude. Chaque section flexible (k) du robot est constituée d'une tige cylindrique flexible sur laquelle un certain nombre de disques rigides sont équidistants fixés (Figure 2.1(b)). Leur mouvement spatial s'effectue à l'aide de trois câbles $l_{i,k}$ ($i = 1,2,3$), disposés à 120° l'un de l'autre. Afin d'établir les relations mathématiques décrivant le mouvement du robot, chaque section flexible est assimilée à une série de robots parallèles, qui

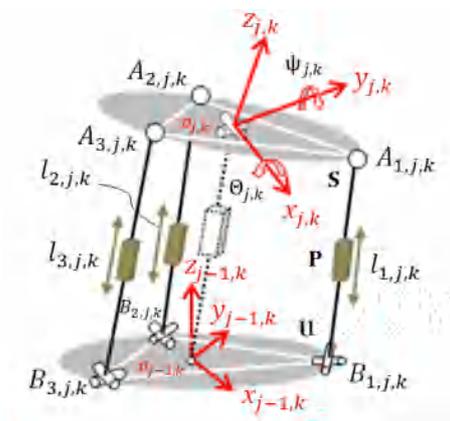
seront appelés unités dans les chapitres suivants, dont son axe central se déforme selon un arc de cercle inextensible [60].



(a)



(b)



(c)

Figure 2.1 : Structure d'un robot continuum actionné par câbles à trois sections flexibles : (a) un robot continuum à trois sections flexibles, (b) une section du robot, et (c) une unité de la section flexible modélisée comme un robot parallèle de type 3UPS-1PU.

La Figure 2.1(c) montre une unité du robot modélisée comme un robot parallèle de type 3UPS-1PU¹. Les actionneurs prismatiques P de la chaîne cinématique UPS sont considérés comme actifs, tandis que celui central de PU est considéré comme passif.

Afin de décrire la configuration du robot, les systèmes de coordonnées suivants sont utilisés (Figure 2.1) :

- Un repère de référence global $\mathfrak{R}_0(O_0X_0 Y_0Z_0)$ attaché à la base de la première section du robot ;
- Un repère de référence local $\mathfrak{R}_k(O_kX_k Y_kZ_k)$, avec $k = 1, 2, \dots, n$, associé à chaque section flexible k ;
- un repère de référence local $\mathfrak{R}_{j,k}(O_{j,k}X_{j,k}Y_{j,k}Z_{j,k})$, avec $j = 1, 2, \dots, m$, associé à chaque à chaque unité j, k .

Comme mentionné précédemment, l'axe central de chaque section flexible est modélisé par un arc de cercle, inextensible et orientée dans l'espace. La première extrémité, supposée fixe dans ce cas, est placée à l'origine de la trame \mathfrak{R}_{k-1} et la deuxième extrémité, attachée au repère \mathfrak{R}_k , peut se trouver à n'importe quel point de l'espace de travail de cette section. Cette section flexible est paramétrée par sa longueur de l'arc ℓ_k , sa courbure κ_k et l'orientation φ_k (Figure 2.1(b)). Pour établir des relations mathématiques entre ces variables et la position et l'orientation de l'extrémité du robot, certaines hypothèses sont nécessaires.

2.3 . Hypothèses de modélisation

- Le robot continuum est décrit par un ensemble de sections flexibles connectées en série.
- Chaque section comporte un nombre fini d'unités identiques reliées en série ;
- Chaque unité est modélisée comme un robot parallèle de type 3UPS-1PU ;
- L'axe central de chaque section flexible est inextensible et se déforme selon au arc de cercle ;
- Les longueurs de câbles d'actionnement sont fragmentées de manière homogène et égale le long de chaque section flexible ;
- la première tangente de chaque section flexion reste toujours colinéaire à l'axe de rotation ;
- L'effet de torsion est négligé ;

¹ P, U et S désignent respectivement articulation prismatique, articulation cardon et articulation sphérique. Généralement, l'articulation cardon et sphérique sont modélisées par deux et trois articulations rotoïdes à axes concourants, respectivement.

- L'effet de la gravité du robot est ignoré ;
- L'effet de frottement entre les câbles et les trous de passage dans les disques est ignoré ;
- Les forces externes agissant sur le robot continuum sont ignorées à l'exception des forces d'actionnement des câbles ;
- Tous les effets dynamiques dus aux forces d'inertie et centrifuges sont ignorés lors de cette analyse ;
- L'élasticité des câbles est ignorée, c'est-à-dire le câble est considéré comme inextensible.
- Chaque section flexible est commandée individuellement ;
- Chaque section flexible à deux degrés de liberté pour un mouvement spatial ;

2.4. Modélisation géométrique directe

La modélisation géométrique directe d'un robot continuum consiste à exprimer la configuration, position et l'orientation, de l'organe terminal du robot en fonction de la variation des longueurs de câbles de chaque section flexible et leurs paramètres géométriques intrinsèques, autrement dit en fonction des variables de ces actionneurs. Formellement, le modèle géométrique direct (MGD) d'un robot s'écrit :

$$\mathbb{X} = f(\mathbb{Q}) \quad (2.1)$$

tel que :

- \mathbb{X} : est le vecteur des variables opérationnels décrivant la position et l'orientation de l'organe terminal du robot ;
- \mathbb{Q} : est le vecteur des variables des actionneurs (la variation des longueurs des câbles pour le cas des robots continuum actionné par câbles).

D'après la littérature, plusieurs méthodes et approches ont été proposées pour résoudre le modèle géométrique direct des robots continuum [60-75]. Dans ce chapitre, l'approche de l'hypothèse à courbure constante [60] a été adoptée pour décrire le modèle géométrique direct du robot continuum actionné par câbles.

Comme présenté sur la Figure 2.2, le concept de l'approche de l'hypothèse à courbure constante se compose de deux parties avec trois espaces, nommément : l'espace des actionneurs, l'espace de configuration et l'espace de travail (espace des tâches), représentant respectivement les longueurs des câbles, les paramètres d'arc de cercle, et la position du point final du robot. La première partie présente la transformation spécifique du robot qui transforme l'espace des actionneurs en espace de configuration. Le second est la transformation indépendante qui transforme l'espace de configuration en espace de tâches.

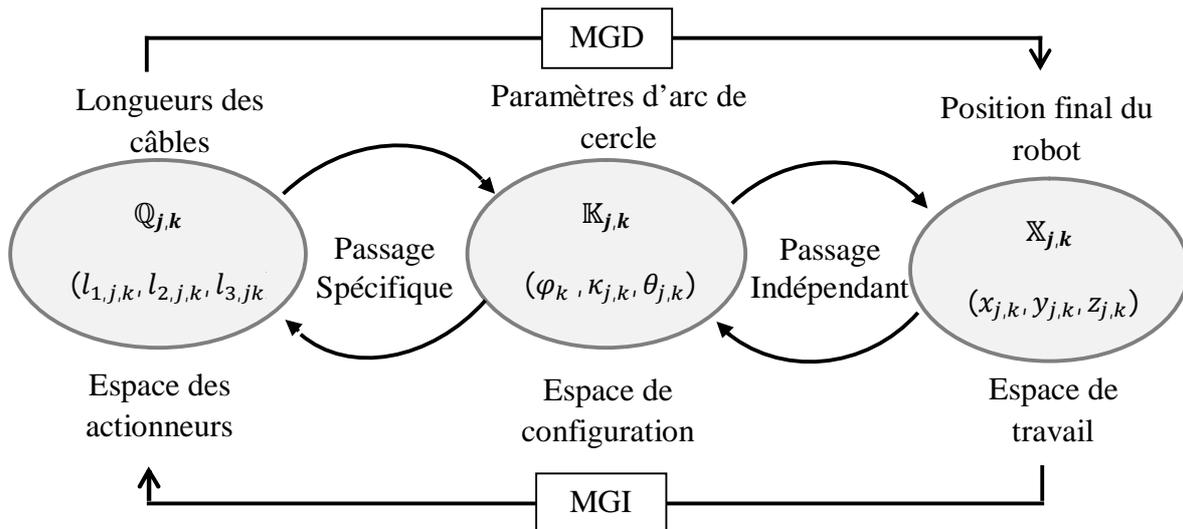


Figure 2.2 : Vue globale de la modélisation géométrique basée sur l'approche de l'hypothèse à courbure constante.

On s'appuyant sur ce qui précède, le modèle de géométrie direct (MGD) d'un robot continuum multi-section peut être dérivé en trois étapes. Premièrement, le calcul du MGD d'une unité. Ensuite la généralisation de ce modèle pour obtenir le MGD d'une section flexible, et enfin la généralisation du MGD de la section flexible pour obtenir le MGD du robot entier. Ces trois étapes sont détaillées dans les sections suivantes :

2.4.1. Modèle géométrique direct d'une unité (j, k)

Afin de modéliser l'unité j de la section k , notée (j, k) , utilisant l'approche de l'hypothèse à courbure constante, le profil de l'ensemble de la structure de l'unité est assimilé à un arc de cercle inextensible orienté dans l'espace. Généralement, ce profil représente l'axe central de l'unité (Figure 2.3).

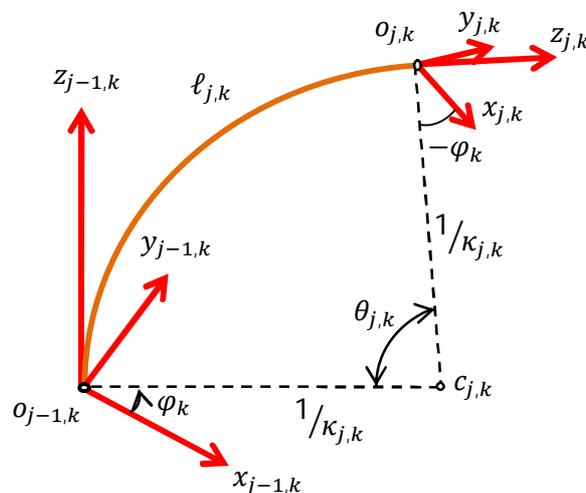


Figure 2.3 : Unité j de la section flexible k modélisée comme un arc de cercle dans l'espace.

Mathématiquement, le MGD d'une unité peut être formellement représenté par l'équation suivante :

$$\mathbb{X} = f_{indep} \left(f_{sp}(\mathbb{Q}) \right) \quad (2.2)$$

Cependant, la représentation standard du MGD est une matrice de transformation homogène ; ainsi, la transformation indépendante, définissant la transformation entre l'espace de configuration et l'espace des tâches, peut être écrite dans un espace tridimensionnel par la matrice suivante :

$$\mathbf{T}_{j,k}^{j-1,k} = \begin{bmatrix} \mathbf{R}_{j,k}^{j-1,k} & \mathbf{P}_{j,k}^{j-1,k} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \quad (2.3)$$

tel que le vecteur position $\mathbf{P}_{j,k}^{j-1,k}$ s'écrit :

$$\mathbf{P}_{j,k}^{j-1,k} = (x_{j,k}, y_{j,k}, z_{j,k})^T \quad (2.4)$$

et la matrice de rotation $\mathbf{R}_{j,k}^{j-1,k}$ écrite sous l'hypothèse à courbure constante peut être obtenue par la multiplication successive de trois matrices de rotation autour des axes principales $z_{j-1,k}$, $y_{j-1,k}$ et $z_{j,k}$ par des angles φ_k , $\theta_{j,k}$ et $-\varphi_k$, respectivement, comme suivant [13, 60, 63] :

$$\mathbf{R}_{j,k}^{j-1,k} = \mathbf{rot}(z_{j-1,k}, \varphi_k) \mathbf{rot}(y_{j-1,k}, \theta_{j,k}) \mathbf{rot}(z_{j,k}, -\varphi_k) \quad (2.5)$$

où $\mathbf{rot}(\cdot, \cdot)$ est la matrice de rotation (3×3).

En développant l'équation (2.5), l'expression finale de la matrice $\mathbf{R}_{j,k}^{j-1,k}$ s'écrit :

$$\begin{aligned} \mathbf{R}_{j,k}^{j-1,k} &= \begin{bmatrix} C\varphi_k & -S\varphi_k & 0 \\ S\varphi_k & C\varphi_k & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C\theta_{j,k} & 0 & S\theta_{j,k} \\ 0 & 1 & 0 \\ -S\theta_{j,k} & 0 & C\theta_{j,k} \end{bmatrix} \begin{bmatrix} C\varphi_k & S\varphi_k & 0 \\ -S\varphi_k & C\varphi_k & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} C^2\varphi_k C\theta_{j,k} + S^2\varphi_k & C\varphi_k S\varphi_k (C\theta_{j,k} - 1) & C\varphi_k S\theta_{j,k} \\ C\varphi_k S\varphi_k (C\theta_{j,k} - 1) & S^2\varphi_k C\theta_{j,k} + C^2\varphi_k & S\varphi_k S\theta_{j,k} \\ -C\varphi_k S\theta_{j,k} & -S\varphi_k S\theta_{j,k} & C\theta_{j,k} \end{bmatrix} \end{aligned} \quad (2.6)$$

tels que :

$$\begin{cases} C\theta_{j,k} = \cos(\theta_{j,k}) \\ C\varphi_k = \cos(\varphi_k) \\ S\theta_{j,k} = \sin(\theta_{j,k}) \\ S\varphi_k = \sin(\varphi_k) \end{cases} \quad (2.7)$$

Pour exprimer les paramètres de l'arc de cercle intervenant dans l'équation (2.6) en fonction des longueurs de câbles (transformation indépendante), la forme de chaque unité (j,k) est prise comme un cylindre de rayon $d_{j,k}$ (Figure 2.4) [13]. D'autre part, comme les câbles d'actionnement sont disposés à 120° l'un de l'autre, les expressions de l'angle de placement des trois câbles ($i = 1, 2, 3$) en fonction de l'angle de rotation φ_k pour un mouvement spatial du robot s'écrit :

$$\gamma_{i,k} = \begin{cases} -\varphi_k & i = 1 \\ \frac{2\pi}{3} - \varphi_k & i = 2 \\ -\frac{2\pi}{3} - \varphi_k & i = 3 \end{cases} \quad (2.8)$$

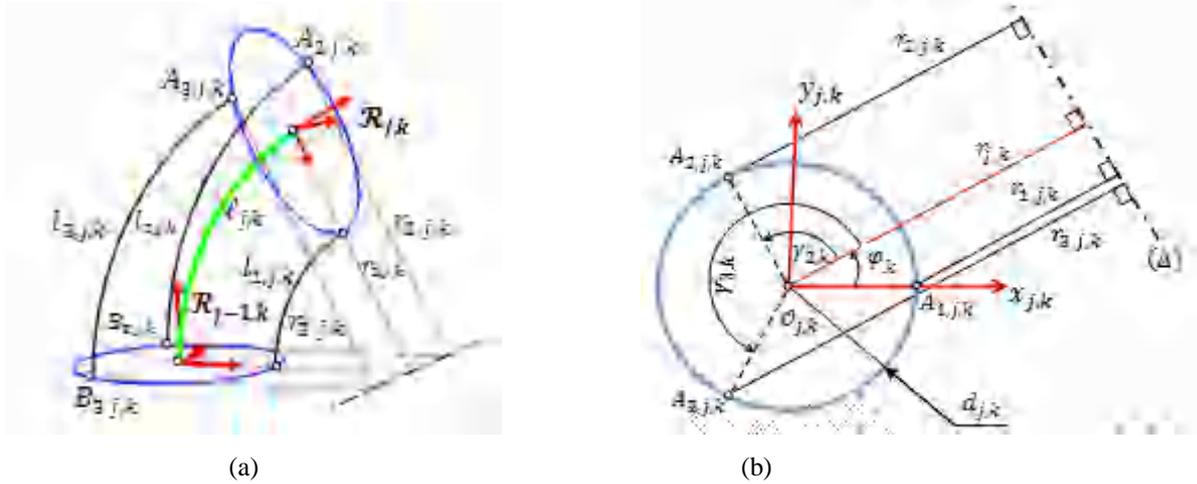


Figure 2.4 : Déformation de l'unité (j,k) selon un arc de cercle: (a) Différentes longueurs et rayons de courbure de l'unité ($\varphi_k = 0$), (b) Différents rayons de courbure de l'unité ($\varphi_k \neq 0$).

A partir de la Figure 2.4 et en utilisant l'équation (2.8), les relations entre les rayons de courbure $r_{i,j,k}$ et les angles de placement des câbles $\gamma_{i,k}$ sont données comme suivant :

$$\begin{cases} r_{1,j,k} = r_{j,k} - d_{j,k}^* \cos(\gamma_{1,k}) \\ r_{2,j,k} = r_{j,k} - d_{j,k}^* \cos(\gamma_{2,k}) \\ r_{3,j,k} = r_{j,k} - d_{j,k}^* \cos(\gamma_{3,k}) \end{cases} \quad (2.9)$$

D'autre part, puisque chaque unité se courbe selon un arc de cercle inextensible, la relation entre la longueur, l'angle de flexion et la courbure de l'unité s'écrit :

$$l_{i,j,k} = \frac{\theta_{j,k}}{\kappa_{i,j,k}} \quad (2.10)$$

En insérant les équations (2.8), (2.9) dans (2.10), les expressions donnant la longueur des câbles d'actionnement en fonction des paramètres de l'arc de cercle s'écrivent :

$$\begin{cases} l_{1,j,k} = \ell_{j,k} - d_{j,k} \theta_{j,k} \cos(\gamma_{1,k}) \\ l_{2,j,k} = \ell_{j,k} - d_{j,k} \theta_{j,k} \cos(\gamma_{2,k}) \\ l_{3,j,k} = \ell_{j,k} - d_{j,k} \theta_{j,k} \cos(\gamma_{3,k}) \end{cases} \quad (2.11)$$

Après simplification de l'équation (2.11) et en utilisant l'équation (2.8), l'expression finale de l'angle de rotation φ_k en fonction des longueurs des câbles s'écrit :

$$\varphi_k = \tan^{-1} \left(\frac{\sqrt{3}(l_{3,j,k} - l_{2,j,k})}{2l_{1,j,k} - l_{2,j,k} - l_{3,j,k}} \right) \quad (2.12)$$

De la même manière et en utilisant la relation trigonométrique de l'équation (2.13), les expressions finale de la courbure $\kappa_{j,k}$ et l'angle de flexion $\theta_{j,k}$ en fonction des longueurs des câbles peuvent être données par les équations (2.14) et (2.15) respectivement.

$$\cos \left(\tan^{-1} \left(\frac{b}{a} \right) \right) = \frac{a}{\sqrt{a^2 + b^2}}, \quad a > 0 \quad (2.13)$$

$$\kappa_{j,k} = \frac{2 \sqrt{l_{1,j,k}^2 + l_{2,j,k}^2 + l_{3,j,k}^2 - l_{1,j,k}l_{2,j,k} - l_{1,j,k}l_{3,j,k} - l_{2,j,k}l_{3,j,k}}}{d_{j,k}(l_{1,j,k} + l_{2,j,k} + l_{3,j,k})} \quad (2.14)$$

$$\theta_{j,k} = \frac{2 \sqrt{l_{1,j,k}^2 + l_{2,j,k}^2 + l_{3,j,k}^2 - l_{1,j,k}l_{2,j,k} - l_{1,j,k}l_{3,j,k} - l_{2,j,k}l_{3,j,k}}}{3d_{j,k}} \quad (2.15)$$

Pour exprimer les coordonnées Cartésiennes $(x_{j,k}, y_{j,k}, z_{j,k})$ de l'extrémité libre de l'unité par rapport à sa base, supposée fixe dans ce cas, en fonction des paramètres d'arc comme suivant (voir Figure 2.3), nous utilisons la transformation indépendante. Cette transformation donne :

$$\begin{cases} x_{j,k} = r_{j,k} C \varphi_k (1 - C \theta_{j,k}) \\ y_{j,k} = r_{j,k} S \varphi_k (1 - C \theta_{j,k}) \\ z_{j,k} = r_{j,k} S \theta_{j,k} \end{cases} \quad (2.16)$$

Finalement, en substituant les équations (2.12), (2.14) et (2.15) dans l'équation (2.16), la transformation géométrique globale, c'est-à-dire le MGD de l'unité (j, k) , est donnée comme dans l'équation suivante :

$$\begin{cases} x_{j,k} = \frac{d_{j,k} l_{\text{somme}}}{2l_{\text{racine}}} \cos\left(\tan^{-1}\left(\frac{2l_{1,j,k} - l_{2,j,k} - l_{3,j,k}}{\sqrt{3}(l_{3,j,k} - l_{2,j,k})}\right)\right) \left(1 - \cos\left(\frac{2l_{\text{racine}}}{3d_{j,k}}\right)\right) \\ y_{j,k} = \frac{d_{j,k} l_{\text{somme}}}{2l_{\text{racine}}} \cos\left(\tan^{-1}\left(\frac{2l_{1,j,k} - l_{2,j,k} - l_{3,j,k}}{\sqrt{3}(l_{3,j,k} - l_{2,j,k})}\right)\right) \left(1 - \cos\left(\frac{2l_{\text{racine}}}{3d_{j,k}}\right)\right) \\ z_{j,k} = \frac{d_{j,k} l_{\text{somme}}}{2l_{\text{racine}}} \sin\left(\frac{2l_{\text{racine}}}{3d_{j,k}}\right) \end{cases} \quad (2.17)$$

tel que :

$$l_{\text{somme}} = (l_{1,j,k} + l_{2,j,k} + l_{3,j,k}) \quad (2.18)$$

$$l_{\text{racine}} = \sqrt{l_{1,j,k}^2 + l_{2,j,k}^2 + l_{3,j,k}^2 - l_{1,j,k}l_{2,j,k} - l_{1,j,k}l_{3,j,k} - l_{2,j,k}l_{3,j,k}} \quad (2.19)$$

2.4.2. Modèle géométrique direct d'une section flexible k

Puisque chaque section flexible k est décrite comme une chaîne cinématique ouverte de m unités, le MGD peut être calculé en multipliant successivement les matrices de transformation élémentaires de toutes les unités, ce qui donne :

$$\mathbf{T}_k^{k-1} = \prod_{j=1}^m \mathbf{T}_{j,k}^{j-1,k} \quad (2.20)$$

tel que $\mathbf{T}_{j,k}^{j-1,k}$ est donnée par l'équation (2.3).

2.4.3. Modèle géométrique direct d'un robot continuum multi-sections

De la même manière, le MGD d'un robot continuum multi-sections peut être calculé en multipliant successivement les matrices de transformation élémentaires de toutes les sections flexible, ce qui donne :

$$\mathbf{T}_n^0 = \prod_{k=1}^n \mathbf{T}_k^{k-1} \quad (2.21)$$

tel que \mathbf{T}_k^{k-1} est donnée par l'équation (2.20).

2.5. Modélisation géométrique inverse

La modélisation géométrique inverse d'un robot continuum consiste à exprimer la variation des longueurs de câbles de chaque section flexible en fonction de position et l'orientation de l'organe terminal du robot ainsi que de leurs paramètres géométriques intrinsèques. Formellement, le modèle géométrique inverse (MGI) d'un robot continuum s'écrit :

$$\mathbb{Q} = f^{-1}(\mathbb{X}) \quad (2.22)$$

Généralement, le modèle géométrique inverse (MGI) d'une unité et celle d'une section flexible sont faciles à obtenir à l'aide des relations géométriques simples. Cependant, du fait de la redondance du robot continuum, le MGI est difficile à obtenir analytiquement, car : pour une configuration opérationnelle donnée de l'extrémité libre du robot, il existe un nombre infini des solutions du MGI. Dans ce cas, deux problèmes se posent : obtenir toutes les solutions du problème du MGI et la résolution de la redondance, c'est-à-dire sélectionner une solution appropriée du MGI qui satisfait un critère spécifique du robot, à savoir : la variation minimale des longueurs de câbles d'actionnement, la distance parcourue par l'extrémité libre du robot, l'évitement d'obstacles, etc. Dans ce qui suit, le MGI d'une unité et celle par d'une section flexible à l'aide de techniques de robots parallèles. Alors que le MGI du robot continuum multi-sections est formulé comme un problème d'optimisation tout en s'appuyant sur les résultats du MGI de l'unité et de la section flexible précédemment développés.

2.5.1. Modèle géométrique inverse d'une unité (j, k)

Comme mentionné précédemment, chaque unité (j, k) de la section flexible est considérée comme étant un robot parallèle de type 3UPS-1PU à deux plateformes, l'une est mobile et l'autre est fixe dans ce cas à deux degrés de liberté (Figure 2.1(c)). Les deux plateformes sont connectées par trois câbles de longueurs ($l_{i,j,k}, i = 1,2,3$). On utilisant les techniques des robots parallèles, les longueurs des câbles d'actionnement peuvent être calculées simplement par la relation suivante :

$$l_{i,j,k} = o_{j-1,k}o_{j,k} + \mathbf{R}_P \cdot o_{j,k}A_{i,j,k} - o_{j-1,k}B_{i,j,k} \quad (2.23)$$

tels que : $A_{i,j,k}$ et $B_{i,j,k}$ sont les points de contact du câble i avec les deux plateformes inférieure et supérieur de l'unité considéré comme un robot parallèle, respectivement. $o_{j-1,k}$ et $o_{j,k}$ sont les centres des deux plateformes autrement dit les originaux des repères $\mathfrak{R}_{j-1,k}$ et $\mathfrak{R}_{j,k}$ respectivement. \mathbf{R}_P est la matrice de rotation (3×3) définissant l'orientation de la plateforme supérieure par rapport à plateforme inférieure. Cette matrice qui peut être exprimée comme suivant :

$$\mathbf{R}_P = \mathbf{rot}(y_{j-1,k}, \Theta_{j,k}) \cdot \mathbf{rot}(x_{j-1,k}, \Psi_{j,k}) \quad (2.24)$$

où $\Theta_{j,k}$ et $\Psi_{j,k}$ représentent les angles de tangage et de roulis correspondant à la rotation autour des axes $y_{j,k}$ et $x_{j,k}$ respectivement. Les expressions de ces angles, en fonction des vecteurs unitaires représentant les éléments de la matrice $\mathbf{R}_{j,k}^{j-1,k}$, peuvent être données comme suit :

$$\Theta_{j,k} = \tan^{-1} \left(\frac{-r_{31}}{r_{11}} \right) \quad (2.25)$$

$$\Psi_{j,k} = \tan^{-1} \left(\frac{-r_{23}}{r_{22}} \right) \quad (2.26)$$

tel que r_{mn} représente l'élément (m, n) de la matrice de rotation $\mathbf{R}_{j,k}^{j-1,k}$ ou ces éléments, c'est-à-dire l'angle de flexion et d'orientation, exprimés en fonction des coordonnées Cartésiennes sont données comme suivants :

$$\theta_{j,k} = \frac{2}{m} \tan^{-1} \left(\frac{\sqrt{x_{j,k}^2 + y_{j,k}^2}}{z_{j,k}} \right) \quad (2.27)$$

$$\varphi_{j,k} = \tan^{-1} \left(\frac{y_{j,k}}{x_{j,k}} \right) \quad (2.28)$$

2.5.2. Modèle géométrique inverse d'une section k

Sous les hypothèses précédentes et du fait que la section flexible k est divisée en m unités identiques, la longueur totale de chaque câble d'actionnement i peut être calculée comme suit:

$$l_{i,k} = \sum_{j=1}^m \sqrt{(l_{i,j,k}^T \cdot l_{i,j,k})} \quad , i = 1,2,3 \quad (2.29)$$

où $l_{i,j,k}$ est la longueur élémentaire de l'unité (j, k) calculée par l'équation (2.23).

2.5.3. Modèle géométrique inverse d'un robot continuum multi-sections

Certes, l'idéal est d'utiliser les méthodes analytiques pour résoudre le problème inverse d'un tel robot continuum, car elles fournissent toutes les solutions possibles pour une pose donnée du robot. Cependant, ces méthodes restent incapables de résoudre ce problème qui est sans doute un problème non trivial. Quant aux méthodes numériques, bien qu'elles fonctionnent indépendamment de la redondance du robot, elles souffrent de nombreuses lacunes, telles que des coûts de calcul élevés, la présence des minimums locaux et l'instabilité des méthodes lorsque la solution est proche des points de singularités. De plus, les critères de décision de ces méthodes sont généralement appliqués pour dériver une solution spécifique et unique. Ainsi, pour comprendre et résoudre les deux problèmes du MGI posés par ce type de robots continuum, un algorithme fiable est développé [65]. Il se compose de deux sous-algorithmes. Le premier sous-algorithme est dédié à fournir un ensemble de solutions possibles pour une position donnée de l'extrémité libre du robot. Autrement dit, il permet de réduire le nombre infini de solutions du MGI à un nombre fini en prenant en compte les

ensembles de poses finales accessibles des premières $(n - 1)$ sections flexibles, qui sont sélectionnés et créés un-par-un de la première section flexible à la $(n - 1)$ ème section, dans laquelle la méthode d'optimisation de l'essaim de particules (PSO) est adoptée pour sélectionner au hasard la première solution du problème. Tandis que le second résout le problème de la redondance, c'est-à-dire pour sélectionner une solution appropriée pour le MGI parmi les variétés existantes qui correspondent parfaitement à la trajectoire ciblée, un critère spécifique a été accordé. La vue globale du schéma fonctionnel de l'algorithme est présentée sur la Figure 2.5 [65].

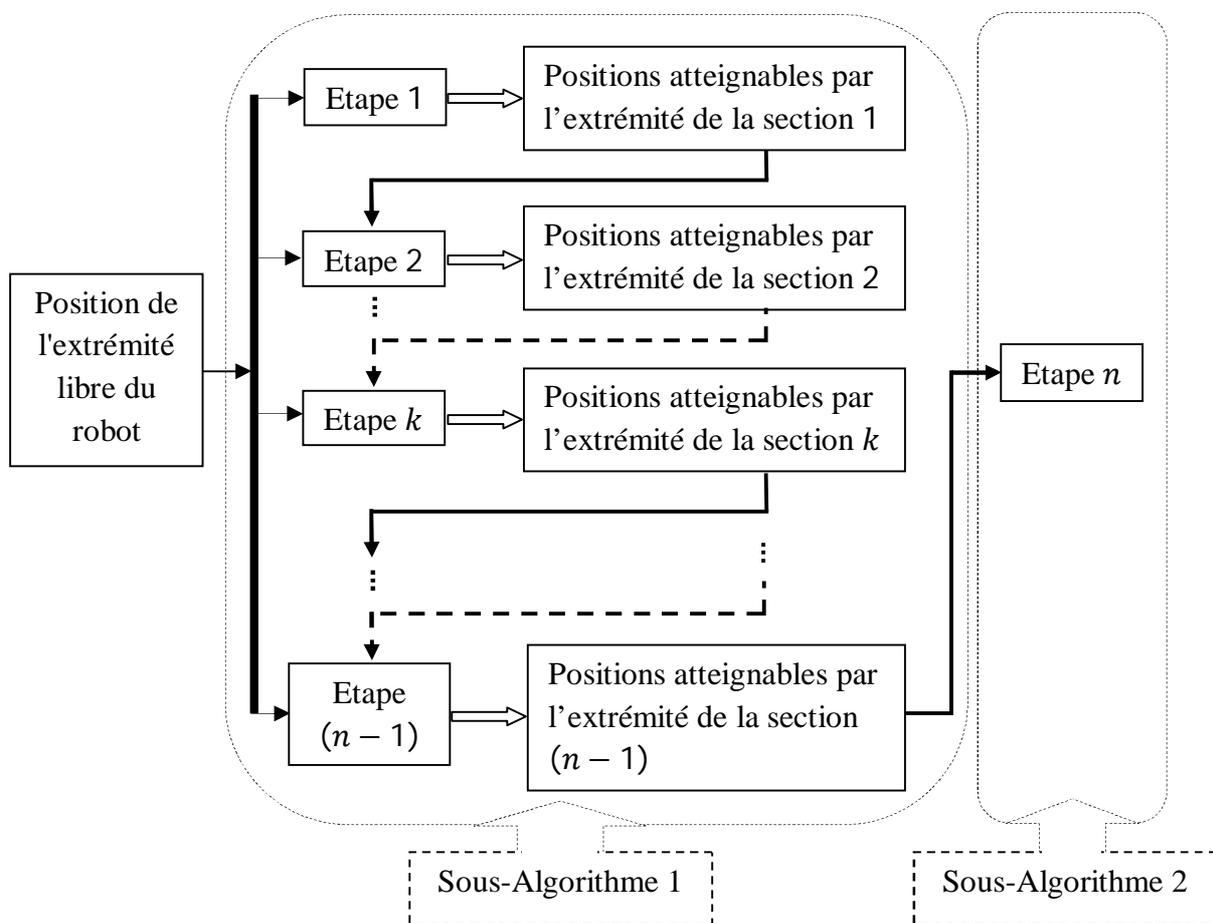


Figure 2.5 : Vue globale du schéma fonctionnel de l'algorithme fournissant des solutions du MGI et la résolution de sa redondance pour un robot continuum à n sections flexibles.

Dans cette partie du chapitre nous ne détaillerons que le premier sous-algorithme, tandis que le deuxième sous-algorithme sera présenté et détaillé dans le chapitre suivant. Cependant, étant donné que l'algorithme proposé est fortement basé sur une méthode d'optimisation, de nombreux algorithmes ou méthodes d'optimisation peuvent être utilisés, à savoir : algorithme génétique (Genetic Algorithm (GA)) [106], optimisation de l'essaim de particules (Particle

Swarm Optimization (PSO)) [107], etc. Bien que ces algorithmes aient réussi à résoudre une variété de problèmes complexes dans différents domaines, chacun d'eux à certaines limites. Cependant, la méthode PSO est simple à coder et converge rapidement par rapport aux autres algorithmes [65, 83], elle sera utilisée pour trouver la première solution aléatoire du MGI pour l'algorithme proposé. Dans ce qui suit, nous présenterons les préliminaires utilisés dans le développement de l'algorithme, à savoir la méthode PS, la fonction objective à minimiser, les conditions aux limites et les limitations physiques du robot continuum.

2.5.3.1. L'algorithme PSO

La méthode PSO standard est une technique de recherche basée sur la population. Elle est développée par Kennedy et Eberhart en 1995 [107]. La méthode est inspirée du comportement social des animaux évoluant en groupe pour chercher la nourriture, tels que les oiseaux, les abeilles et les bancs de poissons.

Le choix de la méthode PSO peut être justifié par sa facilité de mise en œuvre et sa convergence rapide par rapport à d'autres algorithmes. De plus, elle requiert peu de paramètres d'ajustement [65, 83]. Cependant, il ne s'est pas avéré efficace dans certains problèmes difficiles, en particulier ceux où l'algorithme doit être appliqué par petits incréments afin de maintenir la stabilité de l'algorithme. Par conséquent, afin d'augmenter les performances de la méthode, les chercheurs Shi et Eberhart [108] ont ajouté un autre facteur à l'algorithme, appelé poids d'inertie. Ce nouvel algorithme sera appliqué pour trouver la première solution du MGI du robot continuum en question.

Dans l'algorithme PSO, la solution potentielle au problème d'optimisation est appelée une particule et la population de solutions est appelée un essaim. Initialement, les particules sont placées aléatoirement dans l'espace de recherche du problème et chaque particule $p_{\mathcal{D}}$ est caractérisée par un vecteur de position $p_{p,\mathcal{D}}(\xi)$ et un vecteur de vitesse $v_{p,\mathcal{D}}(\xi)$, dont la qualité est évaluée par la valeur de la fonction de coût. Chaque particule mémorise la meilleure position atteinte, notée $pbest_{p,\mathcal{D}}$, tandis que la meilleure position trouvée par toutes les particules de l'essaim est notée $gbest_{\mathcal{D}}$. La vitesse et la position de chaque particule $p_{\mathcal{D}}$ entre les itérations ξ et $\xi + 1$ sont données selon les équations suivantes :

$$v_{p,\mathcal{D}}(\xi + 1) = \omega v_{p,\mathcal{D}}(\xi) + c_1 \lambda_1 (pbest_{p,\mathcal{D}} - p_{p,\mathcal{D}}(\xi)) + c_2 \lambda_2 (gbest_{\mathcal{D}} - p_{p,\mathcal{D}}(\xi)) \quad (2.30)$$

$$p_{p,\mathcal{D}}(\xi + 1) = p_{p,\mathcal{D}}(\xi) + v_{p,\mathcal{D}}(\xi + 1) \quad (2.31)$$

où \mathcal{D} est la dimension du vecteur de particule (dans notre cas : $\mathcal{D} = 3$, c.-à-d. déplacement suivant les trois axes X, Y et Z) ; ω est le facteur du poids d'inertie ; c_1 et c_2 sont deux constantes positives ; λ_1 et λ_2 sont deux nombres aléatoires tirés de l'intervalle $[0, 1]$.

Indépendamment de temps de calcul nécessaire pour résoudre un tel problème, l'obstacle commun à la plupart des algorithmes d'optimisation est la convergence précoce et rapide des solutions vers les minimums locaux, ce qui rend les algorithmes inefficaces dans les applications pratiques. Cependant, pour éviter que l'algorithme ne soit piégé dans des minimums locaux, une technique de régénération de population a été ajoutée à l'algorithme PSO [28, 65]. Sur la base de ce qui précède, les principales étapes de l'algorithme PSO utilisé pour trouver la première solution de MGI sont résumées dans l'algorithme suivant [65] :

Algorithme 1: Pseudo-code de l'algorithme PSO

- 1: **Entrées:** les paramètres du PSO, ξ_{max} .
 - 2: Initialiser les particules aléatoirement et leurs vitesses dans l'espace de recherche \mathcal{D}
 - 3: $\xi \leftarrow 0$
 - 4: **tant que** le critère d'arrêt n'est pas vérifié **faire**
 - 5: **pour** $p_D \leftarrow 1$ à $taille_essaim$ **faire**
 - 6: Evaluer la fonction de coût
 - 7: **fin pour**
 - 8: Sélectionnez $gbest_D$
 - 9: **pour** $p_D \leftarrow 1$ à $taille_essaim$ **faire**
 - 10: Sélectionnez $pbest_{p_D}$
 - 11: Déplacez la particule (p_D) selon les équations (2.30) et (2.31)
 - 12: **fin pour**
 - 13: $\xi \leftarrow \xi + 1$
 - 14: **si** $\xi == \xi_{max}$ **alors**
 - 15: Initialiser les particules aléatoirement et leurs vitesses dans l'espace de recherche \mathcal{D}
 - 16: $\xi \leftarrow 0$
 - 17: **fin si**
 - 18: **fin tant que**
 - 19: **Sortie :** $p_{p,D}$
-

2.5.3.2. La fonction de coût

Étant donné que tout problème d'optimisation nécessite une optimisation de la fonction de coût à minimiser ou à maximiser, la fonction de coût choisie pour le problème en question peut être exprimée simplement comme suit :

$$C_f = \|\mathbb{X}_{désiré} - \mathbb{X}_{génééré}\| \quad (2.32)$$

où $\mathbb{X}_{désiré}$ et $\mathbb{X}_{génééré}$ représentent respectivement le vecteur de position désiré et généré par l'extrémité libre du robot, dont lequel le vecteur de position généré est déterminé par la quatrième colonne de la matrice de transformation indépendante T_n^0 définissant le MGD du robot.

2.5.3.3. Sélection des paramètres de l'algorithme PSO

la performance et l'efficacité de l'algorithme PSO dépendent fortement du choix de ses paramètres, à savoir : les paramètres constants c_1 et c_2 , le poids d'inertie ω , la taille de l'essaim et le nombre d'itérations à effectuer. Pour cette raison, ces paramètres doivent être déterminés soigneusement avant de commencer le processus d'optimisation. Par exemple, les paramètres c_1 , c_2 donnent la tendance aux particules à converger vers la meilleure solution locale ou globale. Généralement, ces deux paramètres sont initialisés de manière aléatoire. Ensuite, en augmentant leurs valeurs d'un pas donné, par exemple 0.1, dans un intervalle [0.1, 3] et en calculant la valeur de la fonction de coût. En conséquence, d'après plusieurs tests, il a été constaté que la valeur 1.2 est la meilleure pour les deux paramètres.

Concernant le poids d'inertie ω , nous avons utilisé la technique développée dans la référence [109]. Ce poids d'inertie dynamique varie linéairement au cours du temps pour dans l'intervalle de 0.9 à 0.4 tout au long de l'implémentation afin d'explorer l'espace de recherche rapidement dans les premières étapes, puis d'exploiter la solution optimale dans les étapes ultérieures. D'autre part, la taille de l'essaim et le nombre d'itérations étant fortement liés à la complexité du problème ; cependant, il n'existe pas des règles précises et cohérentes pour déterminer ces facteurs. Ainsi, après plusieurs tests, nous avons constaté que pour que l'algorithme converge dans un temps de calcul raisonnable, la taille des essais est de 15 essais et le nombre d'itérations à effectuer est égal à 45.

2.5.3.4. Condition aux limites

Étant donné que l'axe central du robot est modélisé par des arcs de cercle connectés en série, deux conditions aux limites sont nécessaires pour effectuer toute analyse géométrique.

La première condition est d'attacher le référentiel de l'épine dorsale du robot (axe central) à la référence de la base, et la deuxième est d'associer l'extrémité libre de l'épine dorsale au point désiré dans la trajectoire prescrite. Mathématiquement, ces conditions peuvent être exprimées comme suit :

$$\begin{cases} \mathbb{X}_0 = \mathbb{X}_{base} \\ \mathbb{X}_n = \mathbb{X}_{désiré} \end{cases} \quad (2.33)$$

2.5.3.5. Limitations physiques

Afin d'assurer le bon fonctionnement du robot, la longueur des câbles d'actionnement doit être limitée. Ces limites physiques peuvent être exprimées comme suit :

$$l_k^{min} \leq l_{i,k} \leq l_k^{max} \quad (2.34)$$

tel que : $l_{i,k}$ est la longueur du câble d'actionnement i de la section k ; l_k^{min} et l_k^{max} sont respectivement la valeur minimale et maximale de la longueur de câble accessible.

2.5.4. Exemples d'application

2.5.4.1. Application 1

Le sous-algorithme 1 du schéma fonctionnel illustré dans la Figure 2.5, destiné à fournir un ensemble de solutions possibles pour une position donnée de l'extrémité libre du robot est détaillé sous dessous pour calculer les solutions possibles du MGI d'un robot continuum à trois sections flexible [65].

Sous-Algorithme 1 : Calcul de toutes les solutions du MGI – Application à un robot continuum à 3 sections flexibles

Étape 1: Calcule des positions des points finals atteignables par la section 1

- 1: **Entrées:** les données géométriques du robot, la position désirée de l'extrémité libre (point final) du robot (P_3)
- 2: Trouver une solution aléatoire du MGI par la méthode PSO /* i.e. $P_k(\theta_k, \varphi_k)$ avec $k = 1, 2$ */
- 3: /* trouver une solution spécifique P_{1s} .i.e. trouver $\theta_{1,max}$ dans le plan de flexion φ_1 (lignes 4 – 13) */
- 4: $J \leftarrow vrai$
- 5: **tant que** ($J == vrai$) **faire**

```

6:       $P_{1s} \leftarrow P_1$ 
7:      Fixer  $\varphi_1$  and  $\theta_1 \leftarrow \theta_1 + \Delta\tilde{\theta}_1$ 
8:      Calculer la nouvelle position du point final  $P_1(X_1, Y_1, Z_1)$  en utilisant
      l'équation (2.16)
9:      Fixer  $P_1$  et vérifier l'existence de  $P_2$  à l'aide de la méthode PSO
10:     si  $P_2$  n'existe pas alors
11:          $J \leftarrow faux$ 
12:     fin si
13: fin tant que
14: Calculer les coefficients de l'équation du plan ( $aX + bY + cZ + d = 0$ ) passant
      par le point  $P_{1s}$  et perpendiculaire à  $\overrightarrow{O_0P_3}$ .
15: Les positions des points finals atteignables de la première section sont tous les
      points qui vérifient l'inégalité :  $aX_1 + bY_1 + cZ_1 + d \geq 0$ 
16: /* Sélection d'un ensemble des positions des points finals [MAT_P1], (lignes: 17-
      25) */
17: [MAT_P1]  $\leftarrow$  []
18: pour  $\theta_1 \leftarrow 0 : \Delta\theta_1 : \theta_{1max}$  faire
19:     pour  $\varphi_1 \leftarrow 0 : \Delta\varphi_1 : 2\pi$  faire
20:         Calculer les coordonnées du point final  $P_1(X_1, Y_1, Z_1)$  en utilisant
         l'équation (2.16)
21:         si ( $aX_1 + bY_1 + cZ_1 + d \geq 0$ ) alors
22:             [MAT_P1]  $\leftarrow$  [MAT_P1; P1]
23:         fin si
24:     fin pour
25: fin pour

```

Etape 2 : Calcul des positions des points finals atteignables par la section 2

```

26: Pour  $q \leftarrow 1 : nb_{P_1}$  faire           /*  $nb_{P_1}$ : le nombre des points finals de la
                                             section 1 [MAT_P1]*/
27:     [MAT_P2,q]  $\leftarrow$  []
28:     Fixer  $P_1$  et trouver un point  $P_2$  en utilisant la méthode PSO
29:     Calculer les coefficients de l'équation du plan ( $aX + bY + cZ + d = 0$ )

```

passant par le point P_2 et perpendiculaire à $\overrightarrow{O_1P_3}^{(*)}$

30: Les positions des points finals atteignables de la deuxième section sont tous les points qui vérifient l'égalité: $aX_2 + bY_2 + cZ_2 + d = 0$

31: /* Sélection d'un ensemble des positions des points finals $[MAT_{P_2,q}]$ */

32: **pour** $\theta_2 \leftarrow 0 : \Delta\theta_2 : \theta_{2max}$ **faire**

33: **pour** $\varphi_2 \leftarrow 0 : \Delta\varphi_2 : 2\pi$ **faire**

34: Calculer les coordonnées du point final $P_2(X_2, Y_2, Z_2)$ en utilisant l'équation (2.16)

35: **si** $|aX_2 + bY_2 + cZ_2 + d| \leq \zeta$ **alors** /* ζ : critère de précision */

36: $[MAT_{P_2,q}] \leftarrow [MAT_{P_2,q}; P_2]$

37: **fin si**

38: **fin pour**

39: **fin pour**

40: **Sortie** : $[MAT_{P_1}]_{(nb_{P_1} \times 3)}$, $[MAT_{P_2,q}]_{(nb_{P_2} \times 3)}$ /* Renvoie $[MAT_{P_2}]$ pour chaque point de (MAT_{P_1}) */

41: **fin pour**

2.5.4.2. Application2

Dans cette application, nous présentons et détaillons les équations mathématiques impliquées dans le sous-algorithme 1 pour calculer le MGI d'un robot continuum à deux tronçons flexibles. Les principales étapes de ce sous-algorithme peuvent être résumées comme suit (Figure 2.6).

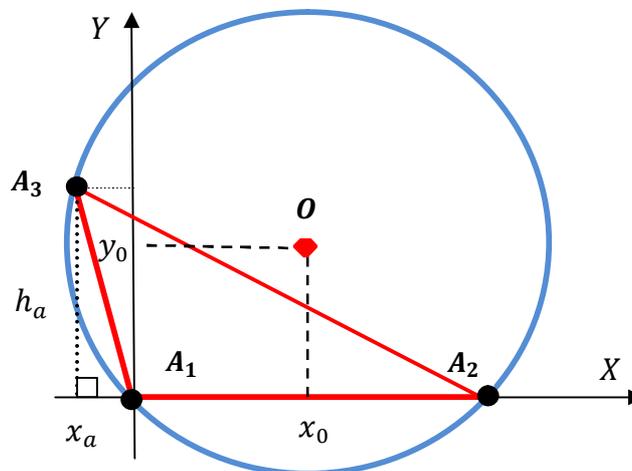


Figure 2.6 : Représentation d'un cercle passant par trois points quelconque dans l'espace.

Sous-Algorithm 1 : Calcul de toutes les solutions du MGI – Application à un robot continuum à 2 sections flexibles

- 1: **Entrées:** la position désiré du point final (P_2)
- 2: Trouver trois solutions aléatoires du MGI en utilisant la méthode PSO, notée $A_j(x_j, y_j, z_j)$, $j = 1, 2, 3$.
- 3: /* Calcul des côtés du triangle A_1A_2 , A_2A_3 et A_1A_3 , notées respectivement a , b et c (voir Figure 2.6)*/
- 4: $a \leftarrow \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}$
- 5: $b \leftarrow \sqrt{(x_2 - x_3)^2 + (y_2 - y_3)^2 + (z_2 - z_3)^2}$
- 6: $c \leftarrow \sqrt{(x_1 - x_3)^2 + (y_1 - y_3)^2 + (z_1 - z_3)^2}$
- 7: /* Calcul de la surface du triangle, notée S */
- 8: $S \leftarrow \frac{1}{4} \sqrt{(a + b + c)(a + b - c)(a - b + c)(-a + b + c)}$
- 9: /* Calcul des coordonnées du point A_3 par rapport au nouveau repère notée x_a et h_a */
- 10: $x_a \leftarrow \frac{a^2 + c^2 - b^2}{2a}$ et $h_a \leftarrow \frac{2S}{a}$
- 11: /* Les coordonnées du centre du cercle passant par A_1, A_2 et A_3 par rapport au nouveau repère sont : */
- 12: $x_0 \leftarrow \frac{a}{2}$, $y_0 \leftarrow \frac{a^2 \times x_a - (x_a^2 + h_a^2)a}{-2a \times h_a}$, $z_0 \leftarrow 1$
- 14: /* Calcul du rayon du cercle */
- 15: $r \leftarrow \sqrt{x_0^2 + y_0^2}$
- 16: /* Les coordonnées des points A_1, A_2, A_3 par rapport au nouveau repère, notées A_{1s}, A_{2s} et A_{3s} sont :*/
- 17: $A_{1s} \leftarrow [0, 0, 1]$; $A_{2s} \leftarrow [a, 0, 1]$; $A_{3s} \leftarrow [x_a, h_a, 1]$;
- 18: /* Calcul de la matrice de transformation (3×3), notée M , définissant le repère $\mathfrak{R}_0(O_0X_0 Y_0Z_0)$ dans le un nouveau repère définit précédemment */
- 19: $L \leftarrow [0 \ 0 \ 1; a \ 0 \ 1; x_a \ h_a \ 1]$
- 20: $N_{11} \leftarrow [x_1 \ 0 \ 1; x_2 \ 0 \ 1; x_3 \ h_a \ 1]$,
 $N_{12} \leftarrow [0 \ x_1 \ 1; a \ x_2 \ 1; x_a \ x_3 \ 1]$,
 $N_{13} \leftarrow [0 \ 0 \ x_1; a \ 0 \ x_2; x_a \ h_a \ x_3]$

```

21:  $N_{21} \leftarrow [y_1 \ 0 \ 1; y_2 \ 0 \ 1; y_3 \ h_a \ 1],$ 
     $N_{22} \leftarrow [0 \ y_1 \ 1; a \ y_2 \ 1; x_a \ y_3 \ 1],$ 
     $N_{23} \leftarrow [0 \ 0 \ y_1; a \ 0 \ y_2; x_a \ h_a \ y_3],$ 
22:  $N_{31} \leftarrow [z_1 \ 0 \ 1; z_2 \ 0 \ 1; z_3 \ h_a \ 1],$ 
     $N_{32} \leftarrow [0 \ z_1 \ 1; a \ z_2 \ 1; x_a \ z_3 \ 1],$ 
     $N_{33} \leftarrow [0 \ 0 \ z_1; a \ 0 \ z_2; x_a \ h_a \ z_3]$ 
23:   pour  $i \leftarrow 1 : 3$  faire
24:     pour  $j \leftarrow 1 : 3$  faire
25:        $M(i, j) \leftarrow \frac{\det(N_{ij})}{\det(L)}$ 
26:     fin pour
27:   fin pour
28:   /* les coordonnées du centre du cercle par rapport au repère de référence*/
29:    $R \leftarrow M \times [x_0 \ y_0 \ 1]'$ 
30:    $M_{GI} \leftarrow [ ]$ 
31:   pour  $\theta \leftarrow 0 : \Delta\theta : 2\pi$  faire
32:      $x_s \leftarrow r \cos(\theta) + x_0$ 
33:      $y_s \leftarrow r \sin(\theta) + y_0$ 
34:      $z_s \leftarrow 1$ 
35:      $V_{xyz} \leftarrow M \times [x_s \ y_s \ z_s]'$ 
36:      $M_{GI} \leftarrow [M_{GI}; V_{xyz}]$  /* Enregistrer le point dans la matrice des solutions du MGI */
37:   fin pour
38:   Sortie:  $M_{GI}$  /* Matrice des solutions du MGI d'un robot continuum à deux
    sections flexibles pour un point donné dans l'espace*/

```

2.6. Simulations numériques

Dans cette section, les modèles géométriques développés sont évalués et vérifiés à travers plusieurs exemples de simulation. Les trois premiers exemples sont dédiés à la détermination de l'espace de travail d'un robot continuum à une, deux et trois sections flexibles en utilisant les expressions des modèles géométriques directs. Afin de vérifier le sous-algorithme 1 consacré pour la détermination des solutions possibles du MGI d'un tel robot continuum, deux autres exemples de simulations sont considérés pour un robot à deux et trois sections flexibles. Les paramètres géométriques décrivant les robots continuum nommément CDCRs

sont présentés dans le Tableau 2.1. Les simulations ont été réalisées dans le logiciel MATLAB sur un ordinateur avec les spécifications suivantes : Intel(R) Xeon(R) CPU E5-1620 0 @ 3,60 GHz, 16 Go de RAM, système d'exploitation 64 bits.

Tableau 2.1 : Paramètres géométriques du CDCR utilisé dans les simulations numériques.

Désignation	Longueur de la section flexible	Distance radiale entre les câbles et la l'axe central (mm)	Rayon des disques (mm)
Section flexible 1	300	40	45
Section flexible 2	250	30	35
Section flexible 3	200	20	25

2.6.1. Détermination de l'espace de travail

Naturellement, la détermination de l'espace de travail d'un tel robot est également très importante dans la phase de conception et de simulation des modèles mathématiques. Par définition, l'espace de travail, également appelé espace de travail maximal, d'un robot continuum est la surface ou le volume d'espace que l'extrémité libre du robot peut atteindre avec au moins une orientation. Dans cette partie, nous présentons l'espace de travail d'un robot continuum à une, deux et trois sections flexibles en utilisant les modèles géométriques directs développés précédemment.

2.6.1.1. Détermination de l'espace de travail d'un robot à une section flexible

Pour construire l'espace de travail d'un robot à une section flexible, il suffit de varier l'angle de flexion dans un certain intervalle pour chaque variation de l'angle d'orientations qui varié également dans un certain intervalle. Par exemple, l'espace de travail accessible pour l'extrémité libre du robot continuum à une section flexible dans les intervalles $\theta \in [0, \pi]$ et $\varphi \in [0, 2\pi]$ est représenté sur la Figure 2.7. Sur cette Figure, les points verts indiquent certaines positions accessibles par l'extrémité libre du robot continuum et la ligne courbée en bleue représente leur axe central, autrement dit l'épine dorsale du robot.

Pour mieux comprendre le comportement du robot continuum multi-sections par rapport à son espace de travail, les Figures 2.8 et 2.9 illustrent certains points atteignables pour chaque section flexible d'un robot continuum à deux et trois sections flexibles respectivement pour un point donné de l'extrémité libre du robot. Dans ces Figures, les épines dorsales de chaque section flexible sont indiquées davantage par les courbes bleue, rouge et verte respectivement.

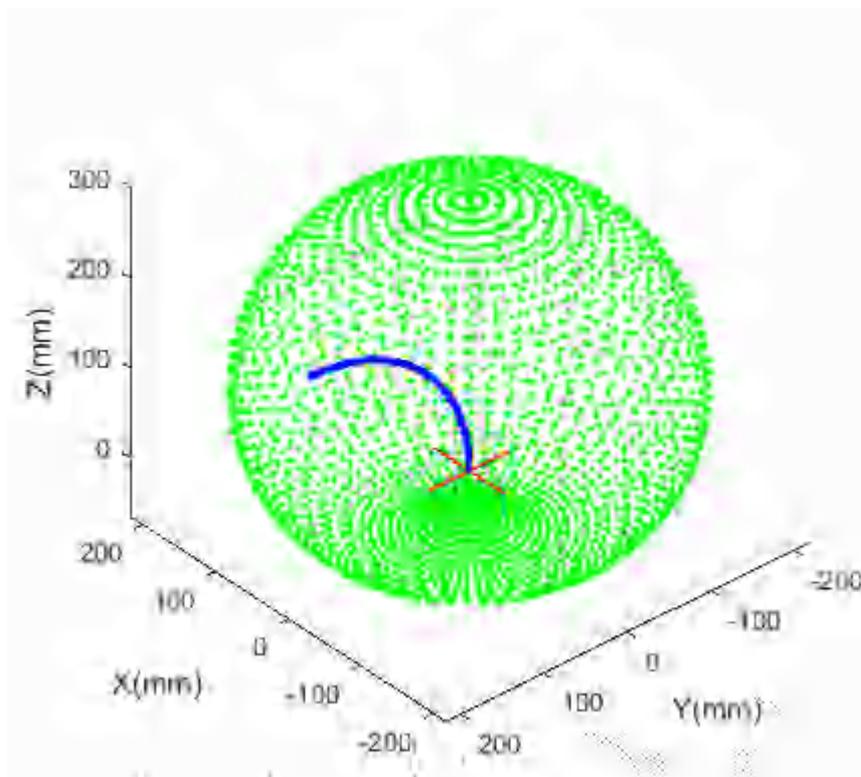


Figure 2.7 : Espace de travail en 3D d'un robot continu à une section flexible.

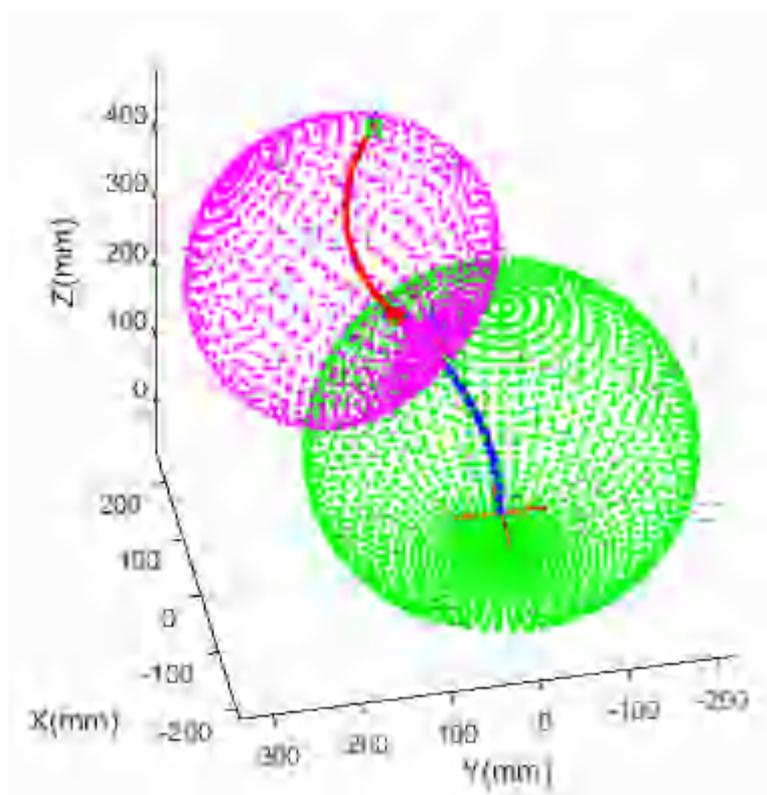


Figure 2.8 : Espace de travail de chaque section flexible d'un robot continu à deux sections flexibles pour un point donné dans l'espace 3D.

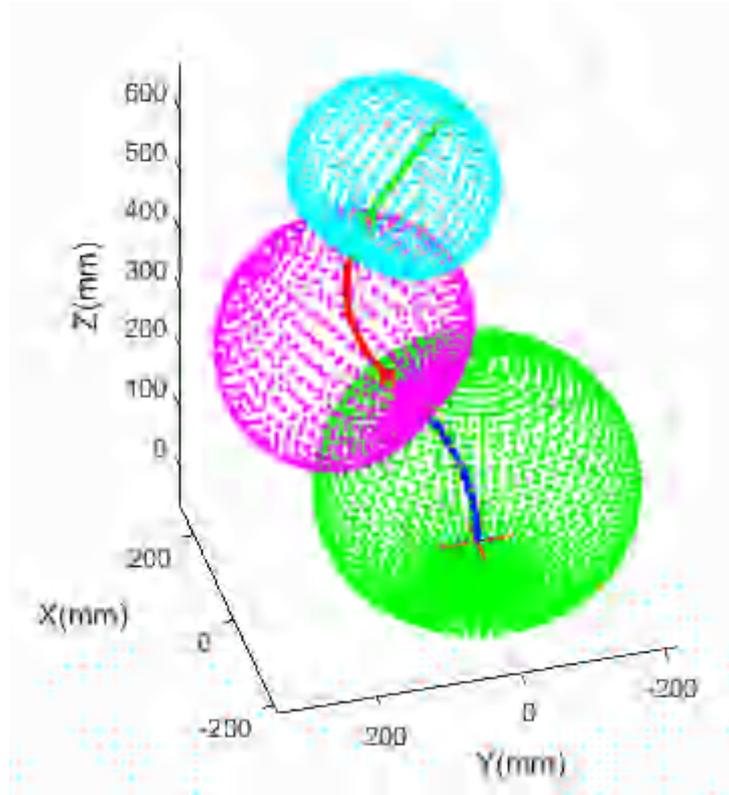


Figure 2.9 : Espace de travail de chaque section flexible d'un robot continuum à trois sections flexibles pour un point donné dans l'espace 3D.

2.6.1.2. Détermination de l'espace de travail d'un robot multi-sections

Les Figures 2.10 et 2.11 représentent quelques positions atteignables (espace de travail) d'un robot continuum à deux et trois sections flexibles respectivement dans lesquelles les épines dorsales de chaque section flexible sont indiquées davantage par les courbes bleue, rouge et verte. Dans la Figure 2.10, la ligne pointée en verte représente les points atteignables par l'extrémité de la première section flexible.

Dans la Figure 2.11, les points atteignables par l'extrémité de la première et la deuxième section flexible sont représentés par les points en magenta et cyan respectivement. Alors que les lignes pointées en noire représentent la trajectoire possible de l'extrémité libre du robot pour des un point donné de l'ensemble de points accessibles de la première et deuxième section flexible.

2.6.2. Simulation des modèles géométriques inverses

Pour vérifier l'efficacité du premier sous-algorithme proposé, qui est consacré au calcul des solutions possibles du MGI pour une pose finale du robot donnée, deux exemples de simulation d'un robot continuum à deux et trois sections flexibles ont été considérés.

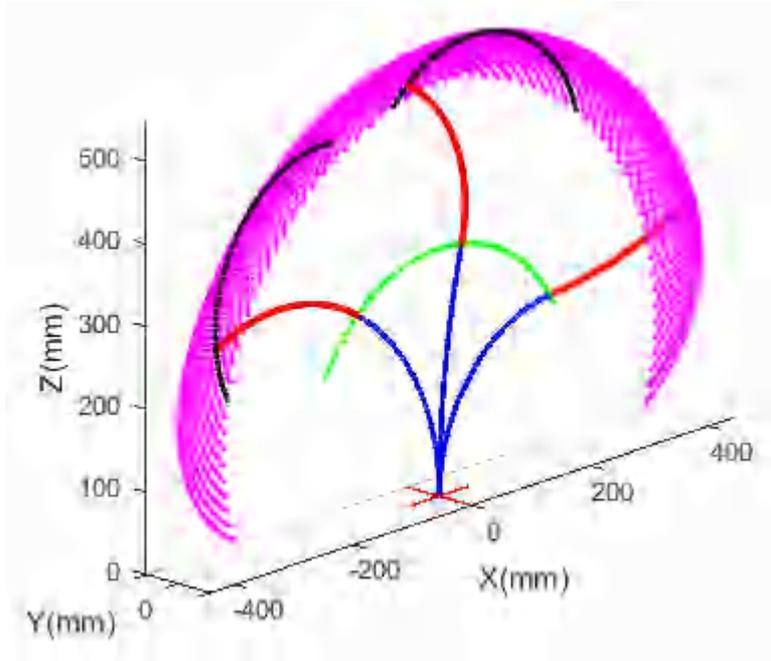


Figure 2.10 : Représentations bidimensionnelle d'un espace de travail d'un robot continuum à deux sections flexibles.

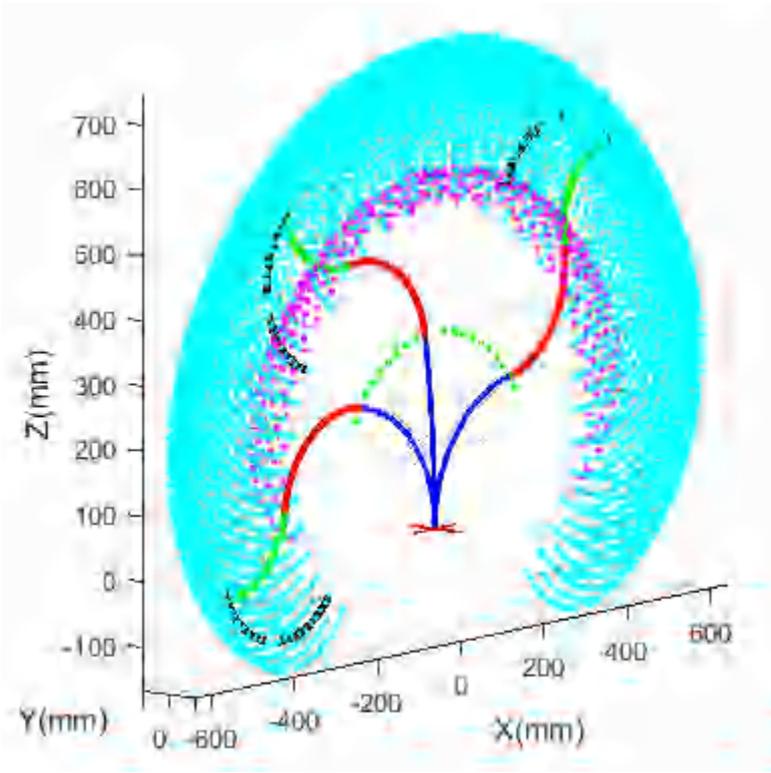


Figure 2.11 : Représentations bidimensionnelle d'un espace de travail d'un robot continuum à trois sections flexibles.

La Figure 2.12 présente les résultats de simulation pour atteindre un point cible d'une trajectoire spirale avec plusieurs solutions du MGI pour un robot continuum à deux sections flexibles. Sur cette Figure, les trois premières solutions aléatoires de MGI obtenues par la méthode PSO sont représentées par des points rouges tandis que le cercle en cyan représente toutes les solutions possibles du MGI.

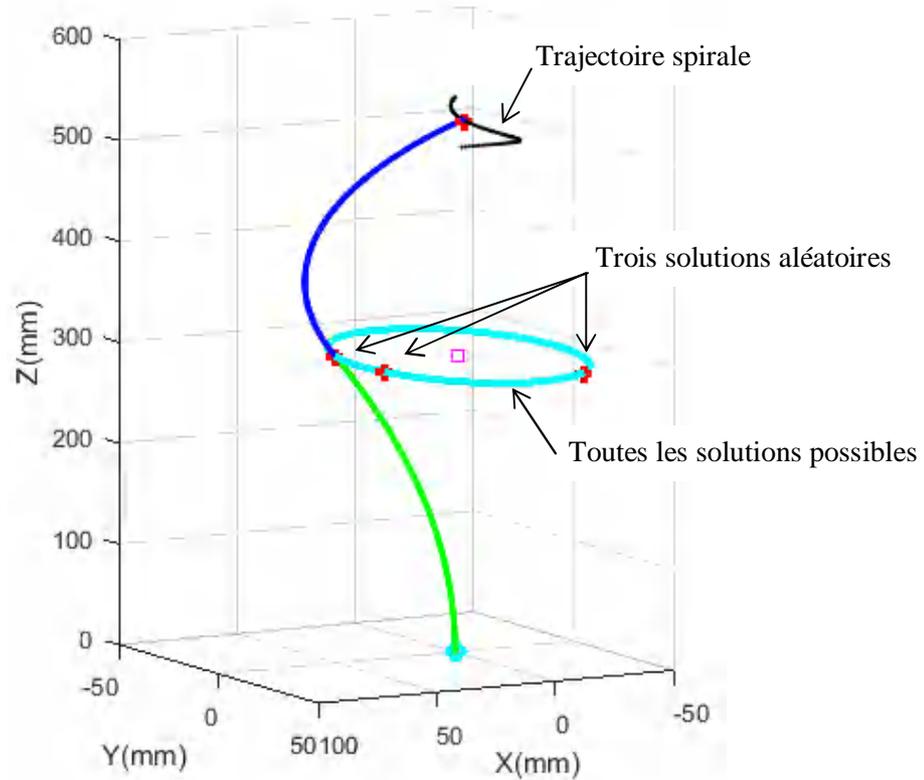


Figure 2.12 : Toutes les solutions possibles du MGI d'un robot continuum à deux sections flexibles pour une pose désirée dans l'espace.

Dans le même contexte, la simulation suivante tente à démontrer davantage l'existence des configurations infinies du robot de n'importe quelle pose de départ à n'importe quelle pose de destination sur la trajectoire désirée pour un robot continuum à trois sections flexibles. D'un point de vue réaliste, il n'est pas possible d'atteindre toutes les solutions du MGI d'un tel robot continuum; cependant, grâce à l'algorithme proposé, le problème peut être réduit à un nombre fini de solutions en considérant les ensembles de poses atteignables des deux premières sections.

Il est à noter que pour un robot à trois sections flexibles, trois cas de singularité peuvent être rencontrés : les deux premières singularités sont lorsqu'une ou deux sections flexibles est

en position droite, et la dernière lorsque toutes les sections flexibles sont en position droite en même temps. Dans les simulations présentées dans ce chapitre et le suivant, ces singularités sont évitées en substituant des valeurs asymptotiques lorsque les angles de flexion sont proches de zéro.

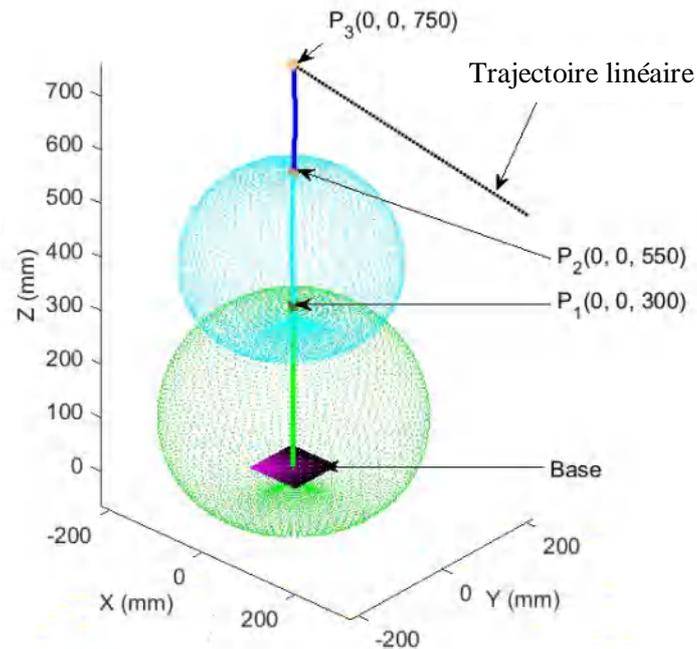


Figure 2.13 : Configuration initiale du robot à trois sections flexibles.

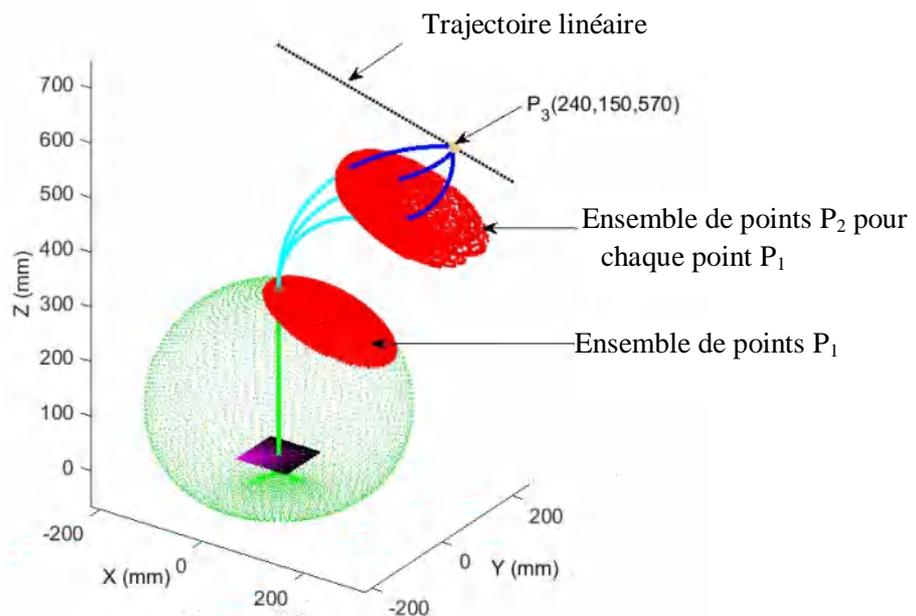


Figure 2.14 : Les points finals atteignables de deux premières sections flexibles pour atteindre un point désiré de la trajectoire.

Avant de déterminer les solutions possibles du MGI du robot considéré, il est évident de noter que lorsque le robot est en ligne droite, il existe une configuration unique du robot, qui représente généralement sa configuration initiale (Figure 2.13), où P_1 et P_2 représentent les poses des deux premières sections flexibles. La Figure 2.14 montre une telle pose de destination du robot dans laquelle les ensembles de poses atteignables par les deux premières sections flexibles sont indiqués par des points rouges. De plus, trois configurations aléatoires du robot sont illustrées davantage dans lesquelles les épines dorsales des sections flexibles sont représentées par les courbes verte, cyan et bleue.

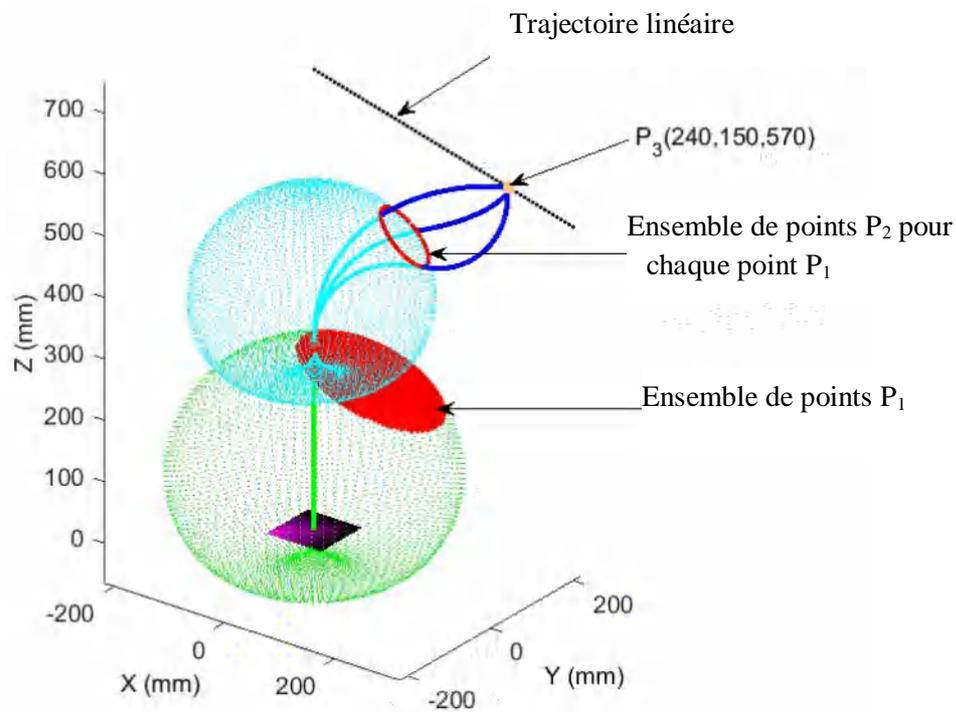


Figure 2.15 : Trois configurations aléatoires du robot pour atteindre un point donné de la trajectoire désirée.

Pour illustrer davantage la Figure 2.14, soit $[MAT_{P_1}] = 4621$ poses pour la première section flexible et $MAT_{P_2, q} = 300$ EP poses pour la deuxième section flexible pour chaque point de $[MAT_{P_1}]$. La dimension du problème devient 4621×300 configurations possibles du robot pour une telle pose sur la trajectoire, ce qui peut être expliqué graphiquement sur la Figure 2.15.

On note ici que l'algorithme proposé permet de déterminer toutes les solutions du MGI pour un robot continuum, alors que pour sélectionner une configuration appropriée du robot l'infinité de solutions se réduit à un nombre fini comme le montre la poursuite de ces

exemples dans le chapitre 3 (c'est-à-dire l'aspect résolution de la redondance le long de la trajectoire souhaitée).

2.6.3. Contributions récentes sur la modélisation géométrique inverse des robots continuum

Du point de vue structural, les robots continuum peuvent être à structure dure ou à structure souple, ce qui rend leurs caractéristiques et propriétés différentes, telles que la taille du robot, le matériau de fabrication, la forme continue/discontinue, etc. D'autre part, ces robots peuvent être contrôlés par différentes entrées, y compris l'actionnement pneumatique, l'actionnement par câble/tendon, l'actionnement hybride (pneumatique + électrique), etc. Sur la base des différences susmentionnées, une comparaison directe entre les modèles mathématiques appliqués n'est possible que sur la base des résultats obtenus. Le Tableau 2.2 liste quelques contributions traitant le problème des robots continuum.

Sans prendre en compte le problème du temps d'exécution de chaque approche, nous pouvons conclure, d'après les résultats du Tableau 2.2, que l'algorithme proposé est le meilleur et peut être utilisé comme une méthode alternative pour résoudre le problème du MGI des robots continuum.

Tableau 2.2 : Quelques contributions traitant le problème du MGI des robots continuum.

Robot	Approche/méthode	Problème du MGI
Robot continuum actionné par câbles (CDCR) : forme continue, structure rigide, contrôlée par câbles.	Approche basée sur PSO [83].	Une seule solution peut être fournie.
Compact Bionic Handling Arm (CBHA): forme continue, structure souple, actionnement pneumatique.	Approche basée sur les courbes de l'hodographe de Pythagore [73].	Une seule solution optimale ou non optimale peut être fournie.
	Approche basée sur les réseaux de neurones [105].	
	Approche hybride : Géométrie et réseau de neurones [82].	
	Approche basée sur SQP [84].	Une seule solution peut être fournie.

	Approche basée sur PSO [83].	
	Approche basé sur la matrice Jacobéenne [86]	
Bionic Handling Arm (BHA): forme continue, structure souple, actionnement pneumatique.	Approche basé sur la matrice Jacobéenne . Goal babbling learning [85].	Une seule solution peut être fournie.
Manipulateur Redondant : forme continue, structure rigide, multi-sections.	Approche Modale [71].	Une seule solution peut être fournie.
Bras continuum : multi-sections, forme continue, structure souple, trois sections, actionnement pneumatique.	Approche Modale [76]. Approche basée sur la géométrie [75].	Une seule solution peut être fournie.

2.7. Conclusion

Dans ce chapitre, nous avons introduit en premier lieu la modélisation géométrique directe d'un robot continuum nommé robot continuum actionné par câbles (en Anglais, Cable-Driven Continuum Robot (CDCR)). Les modèles géométriques directs ont été développés sur la base de l'approche de l'hypothèse de courbure constante. Sur la base de ces modèles, l'espace de travail du robot continuum a été déterminé et un algorithme fiable pour résoudre le problème du Modèle Géométrique Inverse (MGI) a été développé. L'algorithme proposé introduit une nouvelle solution qui aide à réduire le nombre infini de solutions du MGI à un nombre fini de solution en prenant en compte les ensembles de poses atteignables des premières $(n - 1)$ sections flexibles, qui sont sélectionnés et créés un-par-un de la première à la $(n - 1)$ ème section flexible, dans laquelle la méthode d'optimisation de l'essaim de particules (PSO) est adoptée pour sélectionner la première solution aléatoire du MGI. Des simulations numériques ont été réalisées sur un robot continuum câblé à une, deux et trois sections flexibles pointant sur la démonstration des solutions possibles du MGI pour une cible donnée dans l'espace et dans un environnement libre. Les résultats obtenus ont démontré que l'algorithme proposé est précis et efficace sur le plan informatique pour obtenir de toutes les solutions du MGI contrairement aux autres approches proposées dans la littérature. Il est à

noter que l'algorithme proposé peut être appliqué à tout type de robots continuum et quel que soit leur nombre de sections flexibles, ce qui fait de l'algorithme développé un remède souverain au problème du MGI de ces robots continuum.

Chapitre 3

Résolution de la redondance du modèle géométrique inverse

Sommaire

3.1.	Introduction.....	53
3.2.	Critères de résolution de la redondance du MGI d'un robot continuum	53
3.3.	Stratégie d'évitement d'obstacle	54
3.4.	Application	55
3.4.1.	Sous-algorithme 2 utilisant le critère de minimisation du vecteur des variables des actionneurs	55
3.4.2.	Sous-algorithme 2 utilisant le critère des distances parcourues par des extrémités des premières $(n - 1)$ sections flexibles	58
3.5.	Illustration numérique du schéma fonctionnel de l'algorithme proposé	61
3.6.	Simulations numériques	64
3.6.1.	Simulation des modèles géométriques inverses utilisant le sous-Algorithm 2 .	64
3.6.2.	Simulation du modèle géométrique inverse en présence d'obstacle	67
3.6.3	Contributions récentes sur la résolution de redondance du modèle géométrique inverse	69
3.7.	Conclusion.....	70

3.1. Introduction

Dans le chapitre précédent nous avons présenté la première partie (sous-algorithme 1) de l'algorithme proposé pour résoudre le problème du Modèle Géométrique Inverse (MGI) des robots continuum. Il permet de fournir toutes les solutions possibles du MGI pour n'importe quel point de l'extrémité libre du robot. Dans ce chapitre, nous présentons la deuxième partie (sous-algorithme 2) de l'algorithme proposé. Il permet de sélectionner une configuration appropriée du robot satisfaisant un critère donné parmi une grande variété de configurations.

3.2. Critères de résolution de la redondance du MGI d'un robot continuum

Il existe certainement de nombreuses configurations du robot pour une posture désirée, position et orientation, de son extrémité libre, et chaque configuration ou solution est souvent entraînée par des erreurs cartésiennes et d'orientation. Ainsi, pour sélectionner une telle configuration du robot il est plus judicieux d'utiliser le critère approprié dans le processus de résolution de redondance. Dans ce contexte, pour obtenir la configuration optimale du robot dans laquelle son extrémité atteint la cible désirée, les extrémités intermédiaires des premières $(n - 1)$ sections flexibles du robot continuum peuvent être sélectionnées parmi les variétés redondantes qui sont fournies par le sous-algorithme 1 selon un critère spécifique, tel que : minimisation du vecteur des variables des actionneurs, minimisation des distances parcourues par des extrémités des premières $(n - 1)$ sections flexibles, ainsi que les contraintes environnementales affectant le robot, à savoir le cas d'évitement d'obstacles.

– Minimisation du vecteur des variables des actionneurs : Ce critère peut être exprimé comme :

$$l = \min \|l_{som} - l_{som}^{optl}\| \quad (3.1)$$

où l_{som} et l_{som}^{optl} représentent respectivement la somme de toutes les longueurs de câbles d'actionnement pour la configuration actuelle du robot et de la dernière configuration optimale du robot.

– Minimisation des distances parcourues par des extrémités des premières $(n - 1)$ sections flexibles : D'une manière générale, ce critère peut être exprimé comme suivant :

$$dist = \min \sum_{k=1}^{n-1} \|P_k - P_k^{opt}\| \quad (3.2)$$

où P_k et P_k^{opt} représentent respectivement la position actuelle et la dernière position optimale de l'extrémité de chaque section flexible k , pour $k = 1, 2, \dots, n$.

3.3. Stratégie d'évitement d'obstacle

Lorsque le robot travaille dans un environnement restreint, le modèle géométrique inverse consiste à générer une configuration ou posture optimale, parmi une grande variété de configurations, permettant au robot d'éviter les obstacles et d'atteindre la cible désirée. Afin de mettre en œuvre le sous-algorithme 2, qui est décrit ci-dessous, tout en s'appuyant sur la sortie du sous-algorithme 1, la stratégie d'évitement d'obstacles peut être réalisée en deux étapes :

- Premièrement : la sélection d'un ensemble de poses appropriées des premières $(n - 1)$ sections flexibles parmi les variétés de redondance fournies par le sous-algorithme 1, dont lequel le robot n'entre pas en collision avec les obstacles. La stratégie utilisée dans ce travail pour éviter les obstacles consiste à placer les sphères virtuelles de telle sorte que leur centre de gravité soit placé sur l'axe central du robot comme le montre la Figure 3.1.

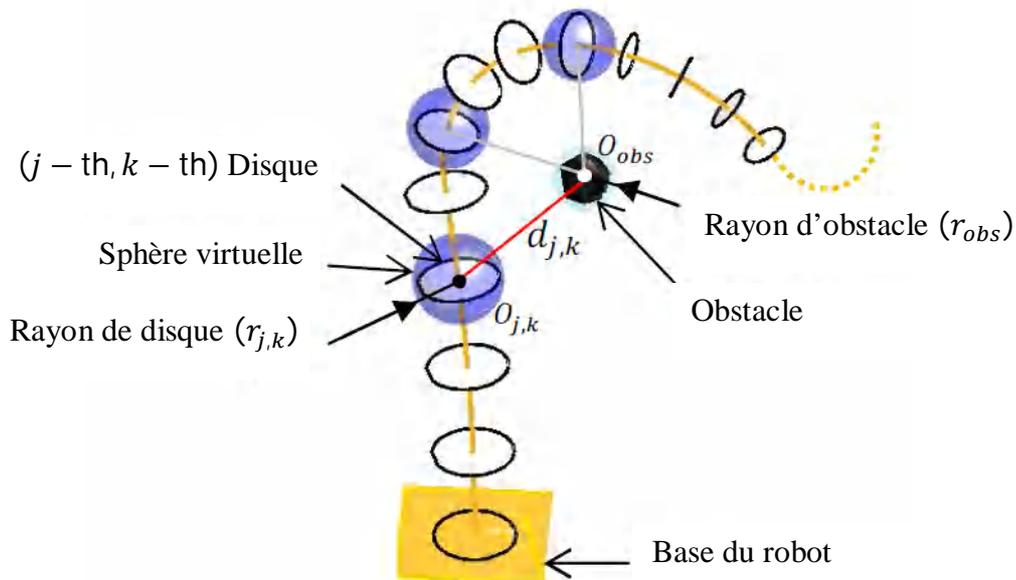


Figure 3.1 : Stratégie d'évitement des obstacles.

- Deuxièmement : si la distance entre un tel obstacle et ces sphères est supérieure à une valeur donnée par rapport à la somme des rayons de l'obstacle et de celle des sphères, la solution du MGI est retenue, sinon la solution est rejetée. Mathématiquement, la stratégie d'évitement d'obstacles peut être formulée comme suit :

$$d_{j,k} > r_{obs} + r_{j,k}, \quad (3.3)$$

tel que : $d_{j,k}$: est la distance entre le centre d'obstacle et le centre de la sphère j de la section flexible k ;

- r_{obs} : est le rayon d'obstacle ;
- $r_{j,k}$: est le rayon de la sphère virtuelle entourant le disque (j, k) .

Pour bien décrire et comprendre l'application de ces critères qui permettent de générer la posture optimale du robot pour une pose donnée de l'extrémité libre du robot, les pseudo-codes représentant le sous-algorithme 2 sont présentés dans la partie suivante :

3.4. Applications

3.4.1. Applications1 : utilisant le critère de minimisation du vecteur des variables des actionneurs

En raison de la complexité de sous-algorithme 2 de l'algorithme proposé, dans cette section, nous l'appliquons à un robot continuum à trois sections flexibles en utilisant le critère de minimisation du vecteur des variables des actionneurs. De plus, pour faciliter la compréhension de l'algorithme, un organigramme illustratif est présenté.

Sous-Algorithme 2 : Génération d'une configuration optimale du robot utilisant le critère de minimisation du vecteur des variables des actionneurs (minimisation des longueurs de câbles d'actionnement) – Application à un robot continuum à 3 sections flexibles

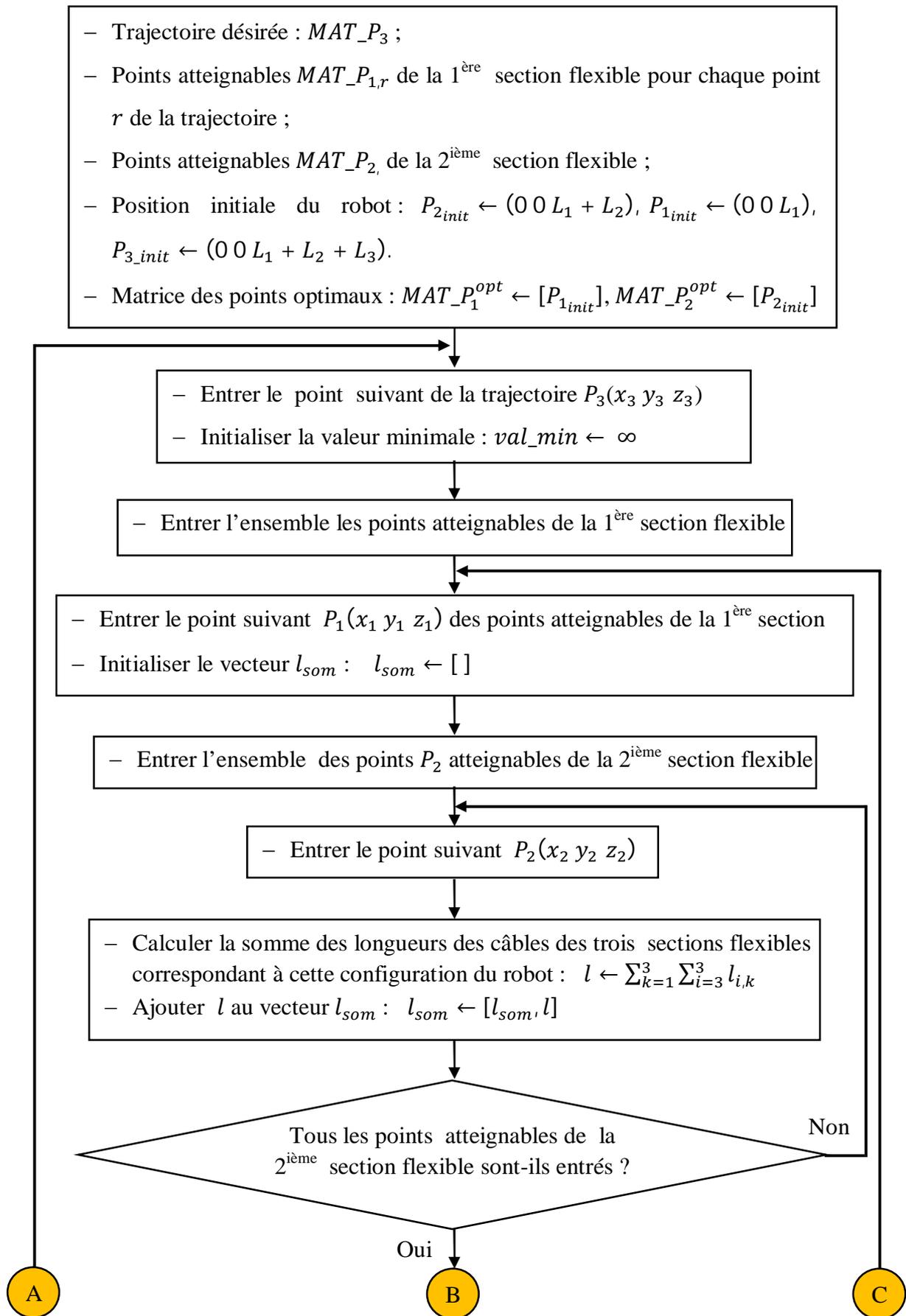
- 1: **Entrées:** Configuration initiale du robot ($P_{1,init}$ et $P_{2,init}$), points finals désirés du $[MAT_P_3]_{(nb_P_3 \times 3)}$, $[MAT_P_2]_{(nb_P_2 \times 3)}$ et $[MAT_P_1]_{(nb_P_1 \times 3)}$ tirés du sous-algorithme 1.
- 2: $[MAT_P_1^{opt}] \leftarrow [P_{1,init}]$ */* Création de la matrice des points optimaux de la première section flexible */*
- 3: $[MAT_P_2^{opt}] \leftarrow [P_{2,init}]$ */* création de la matrice des points optimaux de la deuxième section flexible */*
- 4: $l_{som}^{opt} \leftarrow \sum_{k=1}^3 \sum_{i=3}^3 l_{i,k}$
- 5: **pour** $r \leftarrow 1 : nb_P_3$ **faire**
- 6: $P_3 \leftarrow [MAT_P_3(r, :)]$
- 7: $val_min \leftarrow \infty$
- 8: **pour** $q \leftarrow 1 : nb_P_1$ **faire**

```

9:          $l_{som} \leftarrow []$ 
10:         $P_1 \leftarrow [MAT\_P_1(q, :)]$ 
11:        pour  $u \leftarrow 1 : nb\_P_2$  faire
12:             $P_2 \leftarrow [MAT\_P_{2,q}(u, :)]$ 
13:             $l \leftarrow \sum_{k=1}^3 \sum_{i=3}^3 l_{i,k}$  /* somme des longueurs de câbles selon les
                                                    points  $P_1, P_2, P_3$  en utilisant Eq. (3.1) */
14:             $l_{som} \leftarrow [l_{som}, l]$ 
15:        fin pour
16:         $diff\_l \leftarrow |l_{som}^{opt} - l_{som}|$ 
17:         $[val, rang] \leftarrow \min(diff\_l)$ 
18:         $P_2 \leftarrow [MAT\_P_{2,q}(rang, :)]$ 
19:        si  $val < val\_min$  alors
20:             $val\_min \leftarrow val$ 
21:             $P_1^{opt} \leftarrow P_1$ 
22:             $P_2^{opt} \leftarrow P_2$ 
23:             $l^{opt} \leftarrow \sum_{k=1}^3 \sum_{i=3}^3 l_{i,k}^{opt}$ 
24:             $l_{som}^{opt} \leftarrow l_{som}(rang)$ 
25:        fin si
26:        fin pour
27:         $[MAT\_P_1^{opt}] \leftarrow [MAT\_P_1^{opt}, P_1^{opt}]$ 
28:         $[MAT\_P_2^{opt}] \leftarrow [MAT\_P_2^{opt}, P_2^{opt}]$ 
29:         $l_{som}^{opt} \leftarrow l^{opt}$ 
30:    fin pour
31:    Sortie :  $[MAT\_P_1^{opt}]$  et  $[MAT\_P_2^{opt}]$ 

```

La Figure 3.2 représente un organigramme illustratif du sous-algorithme 2 utilisant le critère de minimisation du vecteur des variables des actionneurs (minimisation des longueurs des câbles des actionneurs) pour générer des configurations optimales du robot à trois sections flexibles.



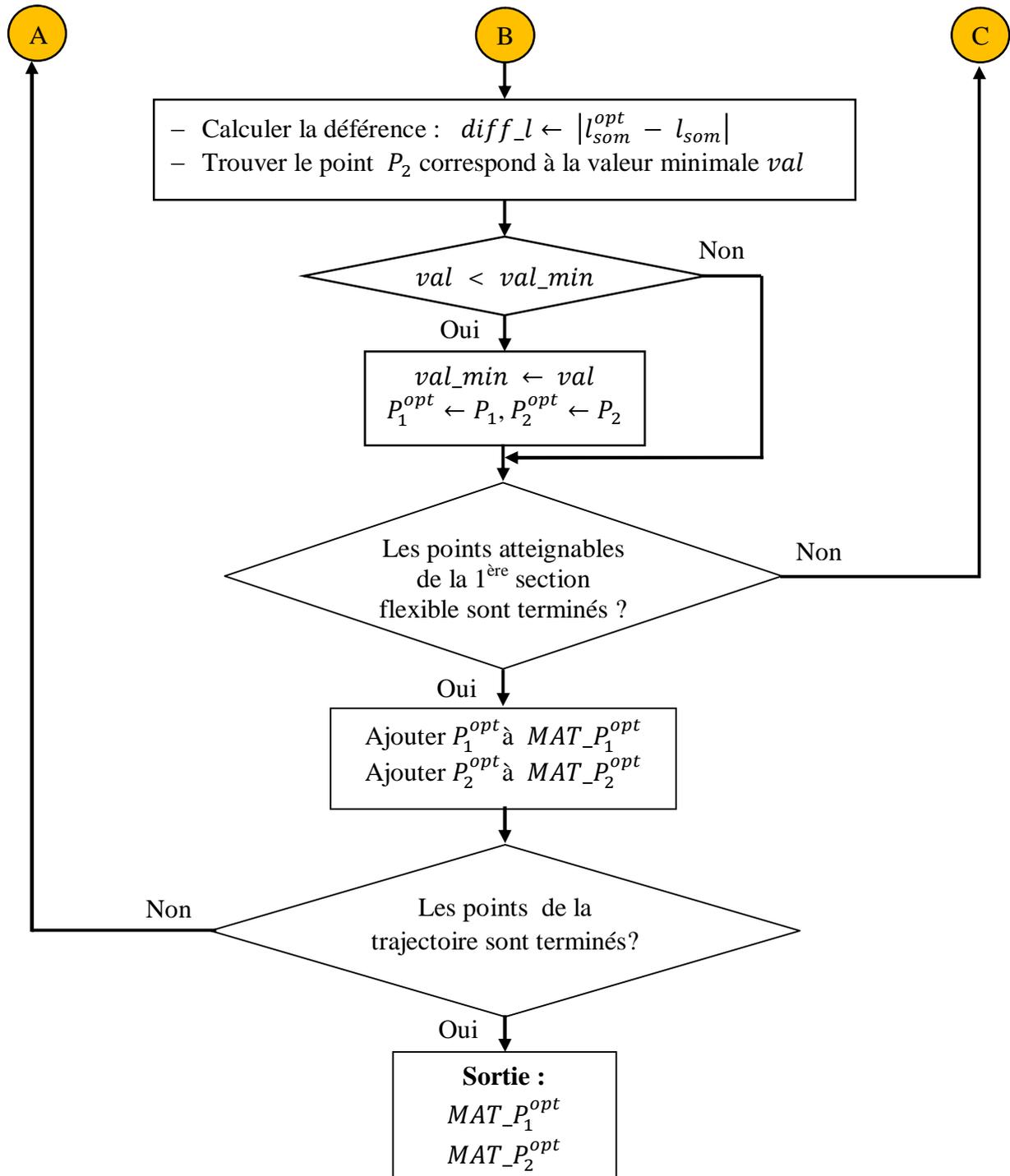


Figure.3.2 : Génération des configurations optimales du robot à trois sections flexibles utilisant le critère de minimisation des longueurs de câbles.

3.4.2. Application2 : Utilisant le critère des distances parcourues par des extrémités des premières $(n - 1)$ sections flexibles

Similaire au critère précédent, dans cette section nous présentons le sous-algorithme 2 utilisant le critère de minimisation des distances parcourues par des extrémités des premières $(n - 1)$ sections flexibles en plus de son organigramme illustratif.

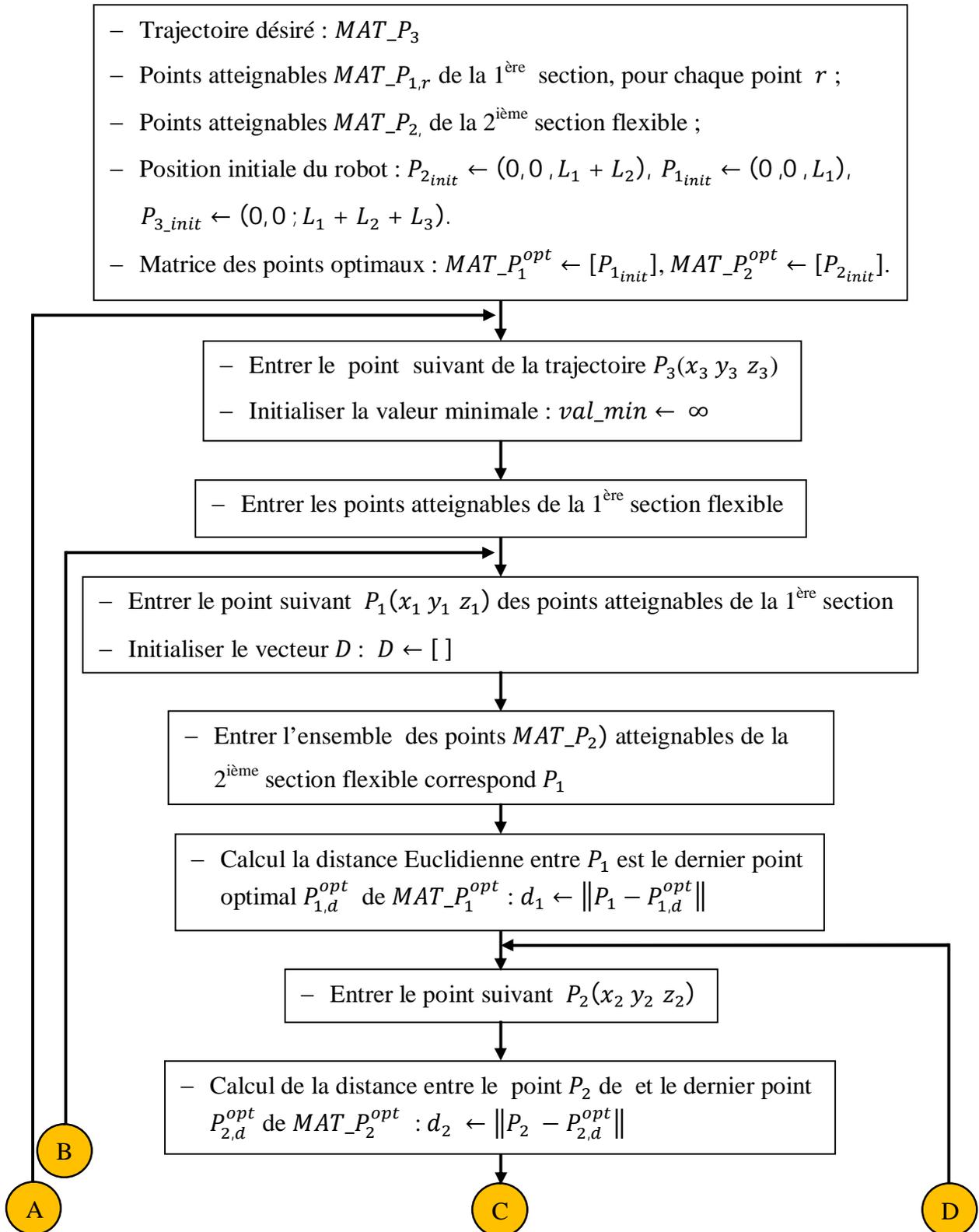
Sous-Algorithmme 2 : Génération d'une configuration optimale du robot utilisant le critère des distances parcourues par des extrémités des premières $(n - 1)$ sections flexibles – Application à un robot continuum à 3 sections flexibles.

```

1:  Entrées: Configuration initiale du robot ( $P_{1,init}$  et  $P_{2,init}$ ), points finals désirés du
      robot  $[MAT\_P_3]_{(nb\_P_3 \times 3)}$ ,  $[MAT\_P_2]_{(nb\_P_2 \times 3)}$ , et  $[MAT\_P_1]_{(nb\_P_1 \times 3)}$ .
2:   $[MAT\_P_1^{opt}] \leftarrow [P_{1,init}]$ 
3:   $[MAT\_P_2^{opt}] \leftarrow [P_{2,init}]$ 
4:  pour  $r \leftarrow 1 : nb\_P_3$  faire
5:       $P_3 \leftarrow [MAT\_P_3(r, :)]$ 
6:       $val\_min \leftarrow \infty$ 
7:      pour  $q \leftarrow 1 : nb\_P_1$  faire
8:           $P_1 \leftarrow [MAT\_P_1(q, :)]$ 
9:           $dist_1 \leftarrow \|P_1 - P_1^{opt}\|$       /* calculer la distance euclidienne entre  $P_1$ 
      actuel et le dernier  $P_1^{opt}$  */
10:          $dist_2 \leftarrow \|(MAT\_P_{2,q}) - P_2^{opt}\|$  /* calculer la distance euclidienne
      entre chaque  $P_2$  actuel de la
      matrice  $[MAT\_P_{2,q}]$  et le dernier
       $P_2^{opt}$  */
11:          $dist \leftarrow dist_1 + dist_2$ 
12:          $[val, rang] \leftarrow \min(dist)$ 
13:          $P_2 \leftarrow [MAT\_P_{2,q}(rang, :)]$ 
14:         si  $val < val\_min$  alors
15:              $val \leftarrow val\_min$ 
16:              $P_1^{opt} \leftarrow P_1$ 
17:              $P_2^{opt} \leftarrow P_2$ 
18:         fin si
19:     fin pour
20:      $[MAT\_P_1^{opt}] \leftarrow [MAT\_P_1^{opt}, P_1^{opt}]$ 
21:      $[MAT\_P_2^{opt}] \leftarrow [MAT\_P_2^{opt}, P_2^{opt}]$ 
22: fin pour
23: Sortie :  $[MAT\_P_1^{opt}]$  et  $[MAT\_P_2^{opt}]$ 

```

La Figure 3.3 représente un organigramme explicatif du sous-algorithme 2 utilisant le critère de minimisation des distances parcourues par des extrémités des premières $(n - 1)$ sections flexibles.



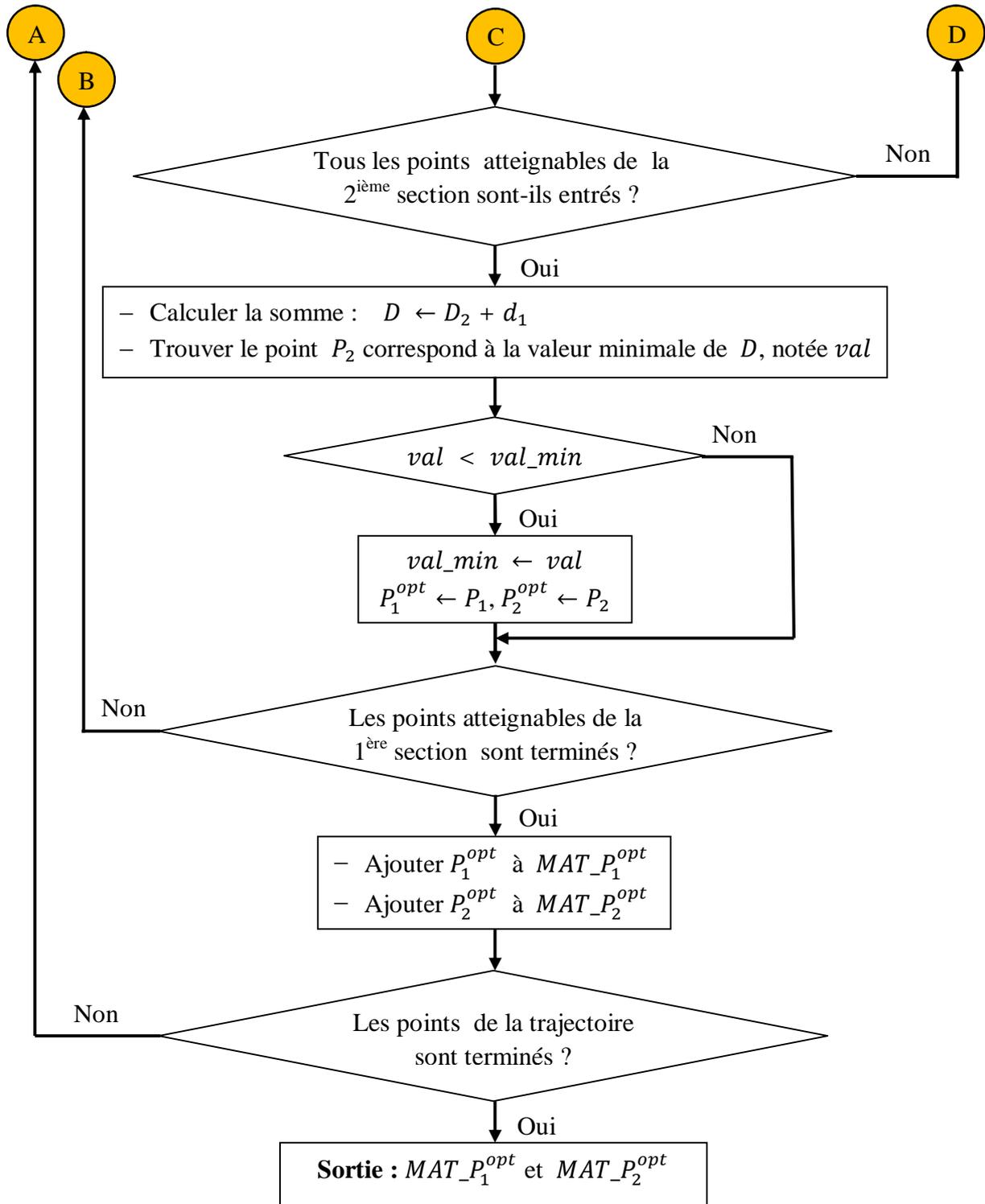


Figure 3.3. Génération des configurations optimales du robot à trois sections le critère de minimisation des distances parcourues par des extrémités des premières $(n - 1)$ sections.

3.5. Illustration numérique du schéma fonctionnel de l'algorithme proposé

Afin d'illustrer en détail le schéma fonctionnel de l'algorithme proposé, tel qu'indiqué sur la Figure 2.5, nous fournirons dans cette section un exemple illustratif démontrant toutes les

étapes de résolution du MGI ainsi que de la redondance. Dans cet exemple, l'algorithme proposé sera appliqué à un robot continuum à trois sections flexibles.

Comme mentionné dans le chapitre précédent, l'algorithme proposé est dédié pour un robot continuum à trois sections flexibles ; ainsi la dimension du problème à l'étape 1 du sous-algorithme 1 est $(nb_{P_1} \times 3)$, où nb_{P_1} est le nombre de poses atteignables de la première section flexible. Alors qu'à l'étape 2, la dimension du problème est $(nb_{P_2} \times 3)$ pour chaque point de $[MAT_{P_1}]$, où $[MAT_{P_1}]$, est la matrice de toutes les solutions de la première section flexible qui vérifient l'inégalité $aX_1 + bY_1 + cZ_1 + d \geq 0$, où X_1, Y_1 et Z_1 sont les coordonnées de la solution spécifique, notée P_{1s} , calculée par la méthode PSO. D'autre part, Les poses accessibles par la deuxième section flexible, notés $[MAT_{P_2}]$, sont tous les points qui vérifient l'égalité $aX_2 + bY_2 + cZ_2 + d = 0$, pour chaque point de $[MAT_{P_1}]$, où X_2, Y_2 et Z_2 sont les coordonnées du point spécifique noté P_2 (étape 2 du sous-algorithme 1). En ce qui concerne la méthode PSO, la population est l'ensemble des poses atteignables possibles et les coordonnées de la particule sont les variables du problème d'optimisation, qui représentent dans notre cas les poses spécifiques, notées P_{1s} et P_2 dans le sous-algorithme 1.

Les paramètres géométriques du robot continuum sont présentés dans le Tableau 2.1. Le robot part à partir de sa position initiale en tant que $[0 ; 0 ; 750]$, voir Figure 3.4(a), pour atteindre la cible souhaitée située en $[240 ; 150 ; 570]$. En fait, il existe une infinité de configurations possibles. Cependant, pour des raisons de simplicité, trois poses sont prises exclusivement pour la première section flexible, notée $[MAT_{P_1}]$; et pour chaque point de $[MAT_{P_1}]$, on choisit quatre poses pour la deuxième section flexible, notée $[MAT_{P_{2,q}}]$, avec $q = 1, 2, 3, 4$. L'ensemble des ces poses forme la matrice de toutes les poses atteignables de la deuxième section flexible, notées $[MAT_{P_2}]$, voir Figure 3.4(b).

Cependant, pour sélectionner une configuration optimale, c'est-à-dire retenir la solution du MGI la plus appropriée, parmi les variétés de solutions envisagées, des critères de minimisation du vecteur des variables des actionneurs et des distances parcourues sont accordés. En conséquence, les poses optimales de la première et de la deuxième section flexible sont données dans le Tableau 3.1 et illustrées graphiquement par des lignes bleues continues et pointillées respectivement sur la Figure 3.4.

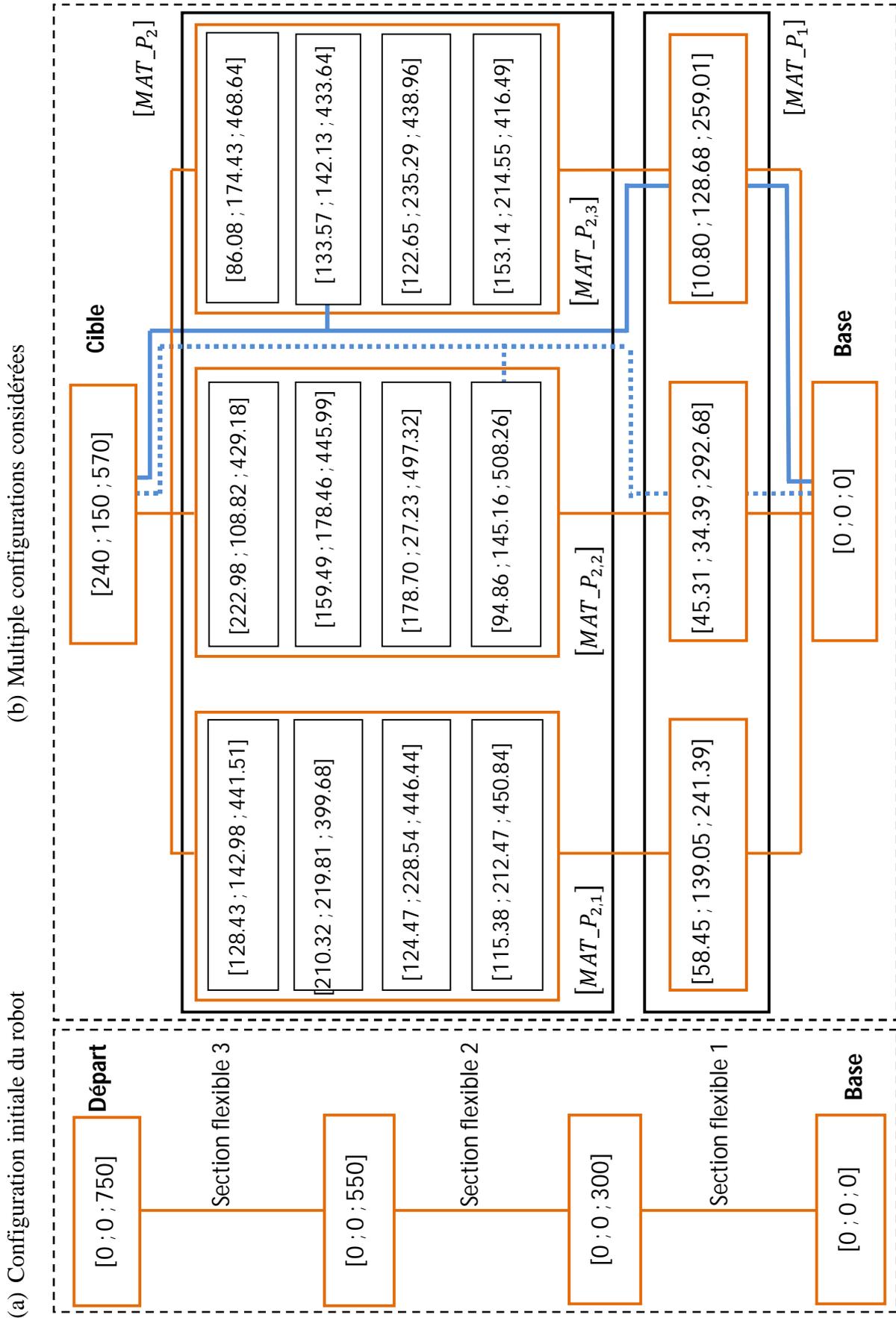


Figure 3.4 : Exemple numérique illustrant le schéma fonctionnel de l'algorithme proposé.

Tableau 3.1 : Poses optimales des extrémités de la première et la deuxième section flexible utilisant les deux critères de minimisation.

Critère	Vecteur des variables des actionneurs	Distances parcourues
Section flexible 1	[10.80 ; 128.68 ; 259.01]	[45.31 ; 34.39 ; 292.68]
Section flexible 2	[133.57 ; 142.13 ; 433.64]	[94.86 ; 145.16 ; 508.26]

3.6. Simulations numériques

Le but de l’algorithme proposé est de calculer toutes les solutions du MGI qui permettent au robot continuum d’atteindre n’importe quelle pose désirée dans son espace de travail (sous-algorithme 1), puis de résoudre le problème de redondance selon un critère donné (sous-algorithme 2). Dans le chapitre précédent, le sous-algorithme 1 a été vérifié à travers plusieurs exemples de simulation ; Cependant, puisque le sous-algorithme 2 dépend fortement des sorties du sous-algorithme 1, dans cette section, nous continuons avec les simulations précédentes pour vérifier le sous-algorithme 2.

3.6.1. Simulation des modèles géométriques inverses utilisant le sous-Algorithme 2

Nous avons vu dans la première simulation du chapitre 2 (voir la section 2.6.2), que toutes les solutions possibles du MGI pour n’importe quel point de trajectoire (Equation (3.4)) peuvent être obtenues en utilisant le sous-algorithme 1 (Figure 2.12). Cette simulation montre les configurations optimales du robot en utilisant le sous-algorithme 2 pour les deux premiers critères précédents (c’est-à-dire minimisation du vecteur des variables des actionneurs et la distance parcourue par l’extrémité de la première section flexible). Les résultats obtenus à partir du suivi de trajectoire sont présentés dans la Figure 3.5. La variation de longueur de câbles pour les deux sections flexibles associées à cet suivi sont données sur la Figure 3.6.

$$\begin{cases} x = 5t \cdot \cos\left(\frac{2\pi}{10}\right) \\ y = 5t \cdot \sin\left(\frac{2\pi}{10}\right) \\ z = 550 - 10t \end{cases} \quad (3.4)$$

Pour mettre en évidence davantage sur l’aspect de la résolution de redondance le long de la trajectoire souhaitée, deux cas sont pris en compte. L’équation mathématique de la trajectoire à suivre est donnée par l’équation suivante. Nous notons ici que cet exemple est une continuation de l’exemple présenté dans la section 2.6.2 consacré au robot continuum à trois sections flexibles.

$$\begin{cases} x = 40t \\ y = 25t \\ z = 750 - 30t \end{cases} \quad (3.5)$$

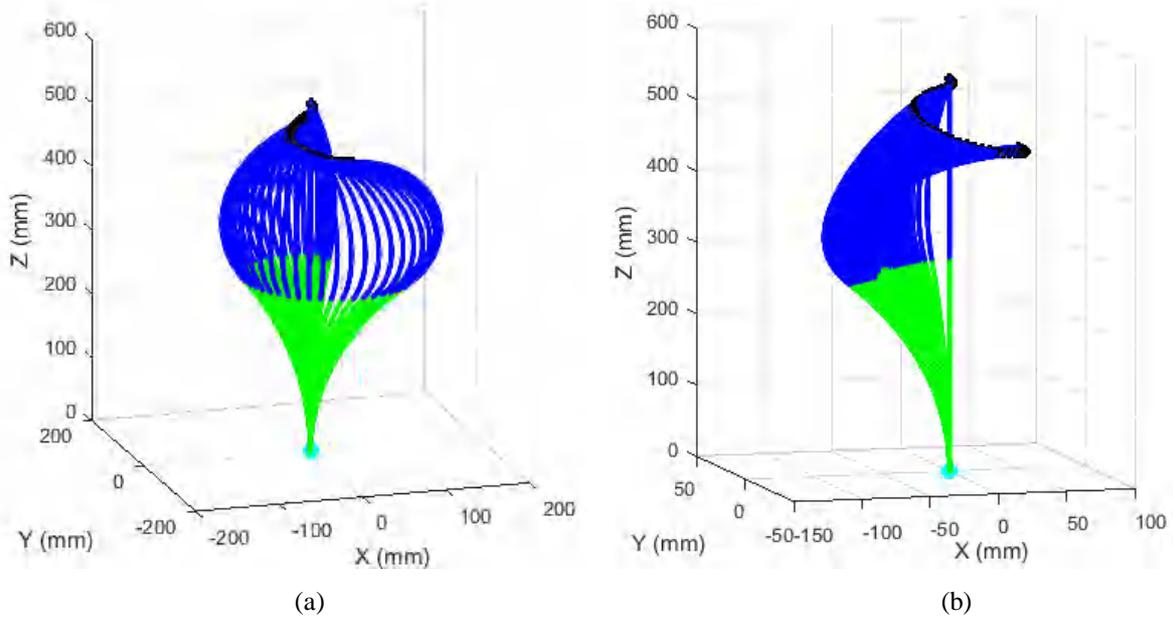


Figure 3.5 : Configurations optimales du robot (a) 1^{er} critère, (b) 2^{ème} critère.

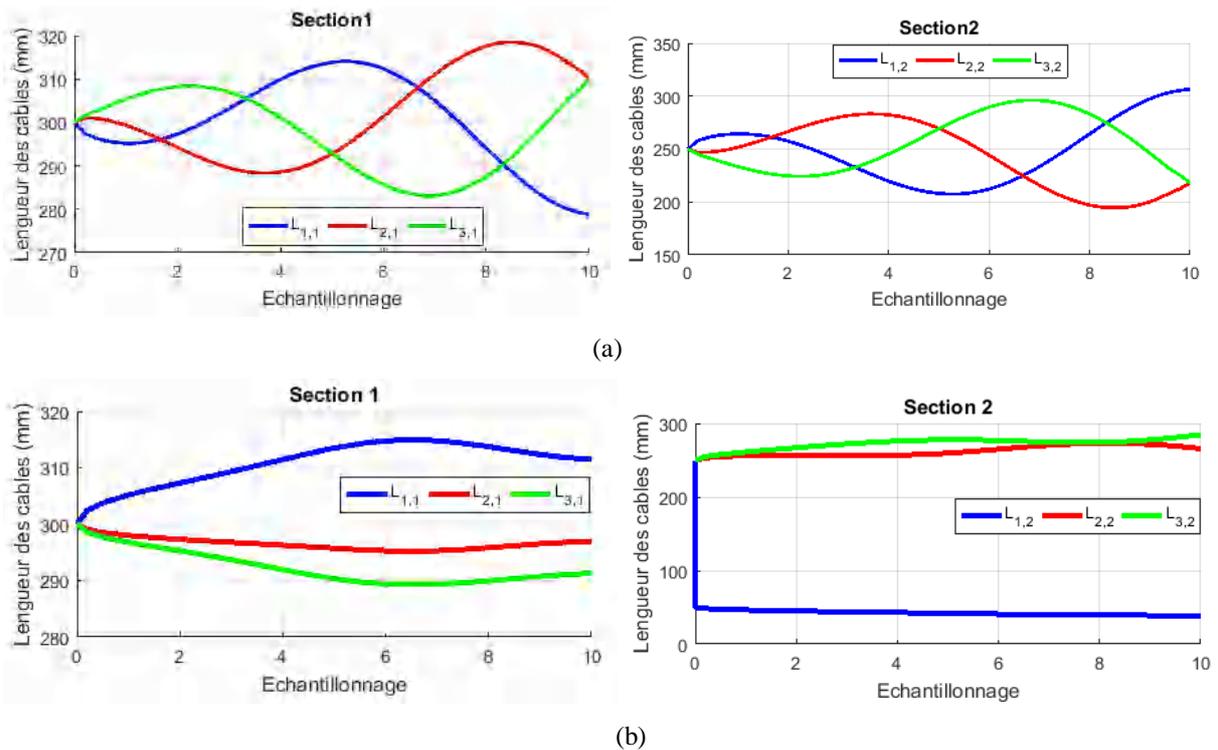


Figure 3.6 Profils des longueurs de câbles nécessaires pour suivre la trajectoire spirale pour les deux critères.

La Figure 3.7 montre les configurations optimales du robot représentées par son épine dorsale (axe central du robot) dans lequel son extrémité libre suit la trajectoire linéaire définie par l'équation (3.5) en utilisant les deux critères précédents (c'est-à-dire minimisation du vecteur des variables des actionneurs et la distance parcourue, respectivement).

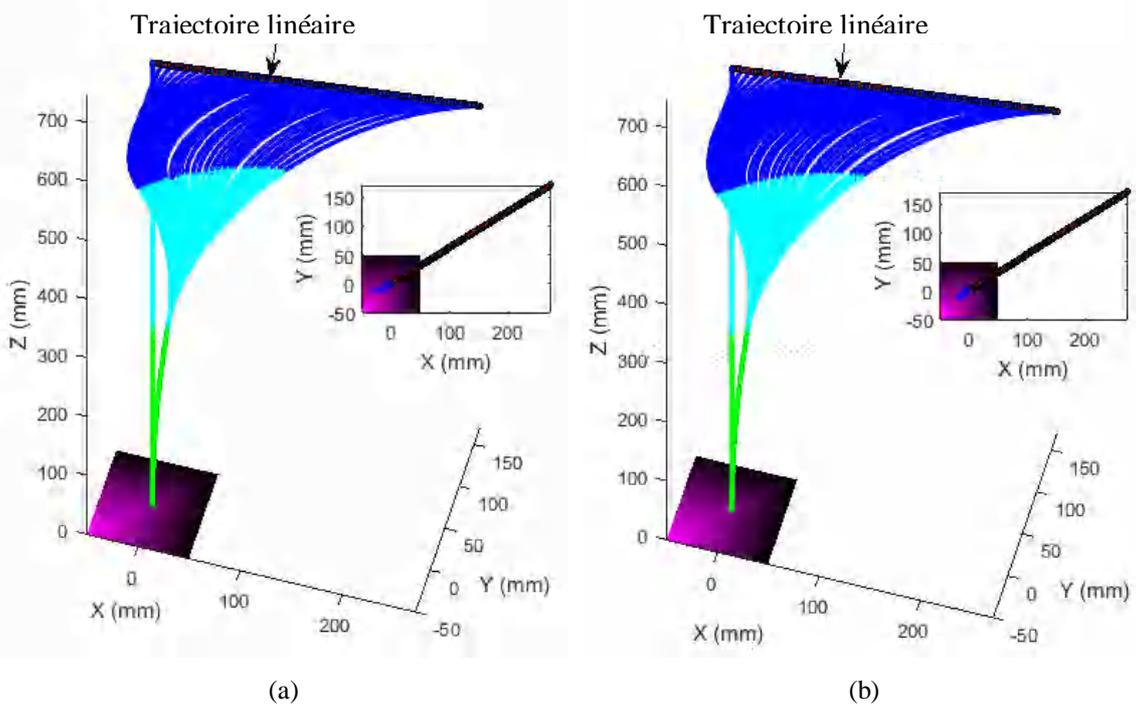


Figure 3.7 : Configurations optimales du robot à trois sections flexibles pour un suivi d'une trajectoire linéaire. (a) 1^{er} critère (minimisation du vecteur des variables des actionneurs), (b) 2^{ème} critère (minimisation des distances parcourues).

La Figure 3.8 montre les profils de longueur des câbles nécessaires pour suivre la trajectoire linéaire avec les deux critères.

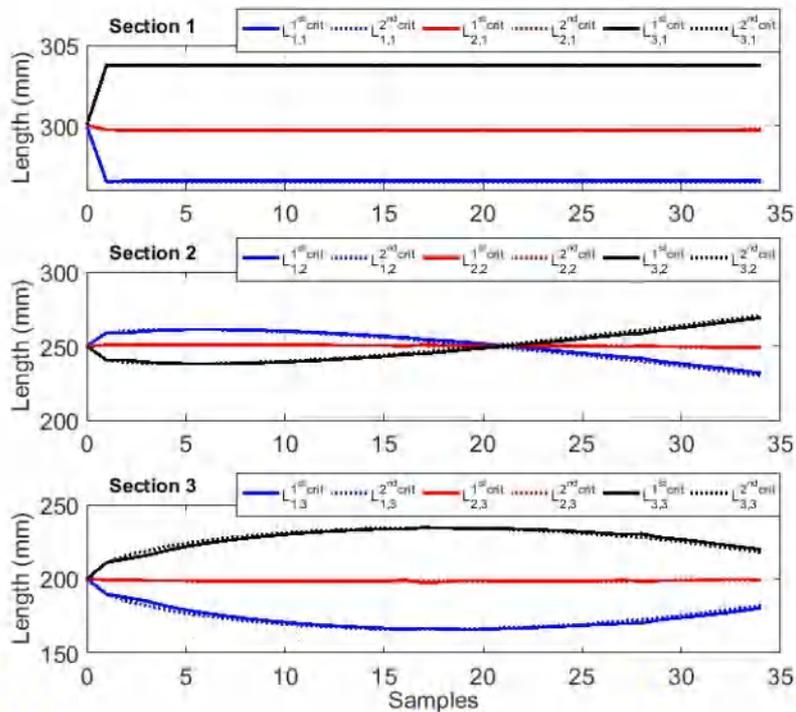


Figure 3.8 : Profils des longueurs de câbles nécessaires pour suivre la trajectoire linéaire par les deux critères.

D'après cette Figure, on peut voir que les deux critères donnent des solutions très similaires pour trouver les solutions du MGI. Tout à fait, les deux solutions sont très similaires car pratiquement et selon la trajectoire donnée, c'est la solution unique et optimale qui peut être obtenue. Ce résultat indique l'efficacité de calcul de l'algorithme proposé pour trouver les solutions du MGI et la résolution de sa redondance. Cependant, la légère différence apparente dans la variation des longueurs, en particulier pour la deuxième et troisième section flexible, est due au petit nombre de poses pris en compte pour la deuxième section flexible lors de la résolution de la redondance.

3.6.2. Simulation du modèle géométrique inverse en présence d'obstacle

Dans cette simulation, l'algorithme proposé est évalué en évitant les obstacles qui peuvent être rencontrés dans l'espace de travail du robot.

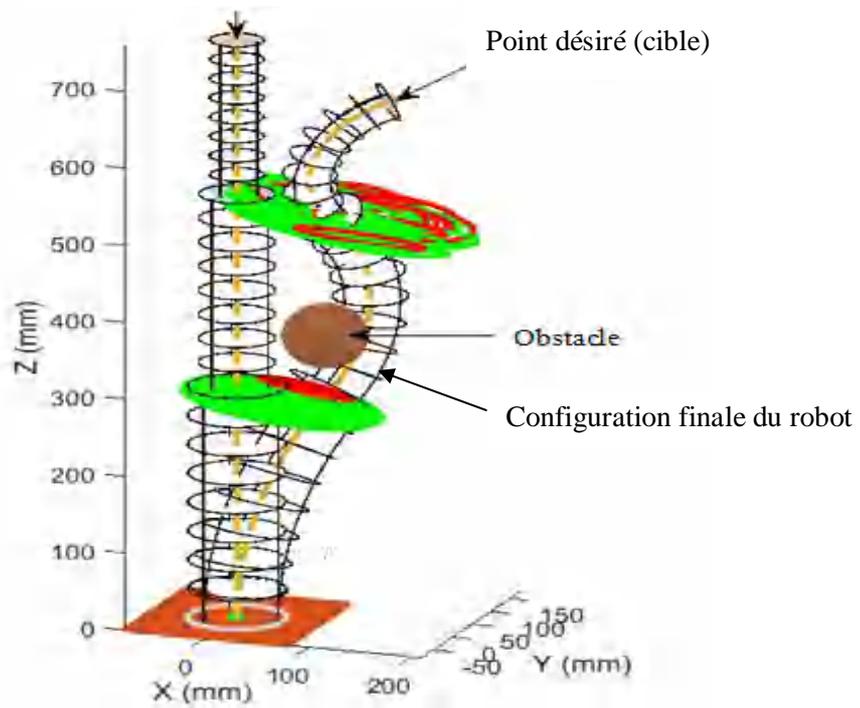
Considérons un obstacle statique situé à $\mathbb{X}_{obs} = [60 ; 70 ; 350]$ (mm), dans l'espace de travail du robot, par rapport au repère de référence fixe. Le robot part de son état initial donné comme $[0 ; 0 ; 750]$ (mm) pour atteindre le but désiré $\mathbb{X}_{désiré} = [240 ; 150 ; 570]$ (mm). Nous notons ici que pour s'assurer que les obstacles le long de la trajectoire sont évités, la trajectoire à suivre doit être subdivisé en un grand nombre de points. Pour atteindre cet objectif, l'évitement d'obstacles est prévu au point de destination.

La Figure 3.9 montre les zones de collision et d'évitement d'obstacles où les points rouges représentent les zones de collision et les points verts représentent les zones d'évitement d'obstacle avec le robot.

Pour éviter cet obstacle, une contrainte supplémentaire a été ajoutée au problème d'optimisation visant à maintenir la position de l'obstacle et certains points de l'axe central du robot à une distance euclidienne d'au moins 40 mm. Cependant, il existe une infinité de configurations possibles pour atteindre le but tout en évitant l'obstacle. Ainsi, pour sélectionner la configuration optimale dans laquelle le robot n'entre pas en collision avec l'obstacle et suit correctement la trajectoire souhaitée, les critères précédents, à savoir : minimisation du vecteur des variables des actionneurs et minimisation des distances parcourues, sont accordés.

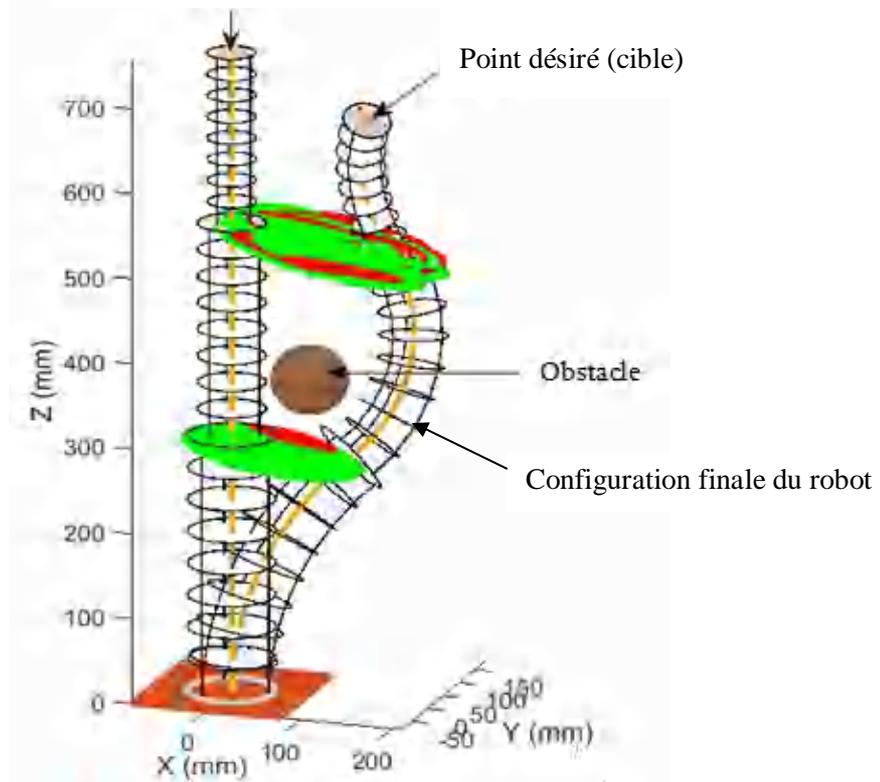
Les poses atteignables des deux premières sections flexibles du robot qui satisfont aux deux critères susmentionnés sont présentées dans le Tableau 3.2. Par ailleurs, les vecteurs des variables des actionneurs (longueurs des câbles) nécessaires pour atteindre le but souhaité à travers les deux critères sont présentés dans le Tableau 3.3.

Configuration initiale du robot



(a)

Configuration initiale du robot



(b)

Figure 3.9 : Configurations de collision et d'évitement d'obstacle du robot (les points rouges sont des zones de collisions, et verts sont des zones libres) : (a) collision, (b) évitement d'obstacle.

Tableau 3.2 : Points optimaux des extrémités de la première et deuxième de section flexible satisfont les deux critères tout en évitant l'obstacle.

Critère	Vecteur des variables des actionneurs	Distances parcourues
Section flexible 1	[145.70 ; 115.48 ; 199.38]	[0 ; 0 ; 300]
Section flexible 2	[258.80 ; 188.08 ; 374.68]	[74.49 ; 46.72 ; 529.53]

Tableau 3.3 : les longueurs de câbles nécessaires satisfont les deux critères tout en évitant l'obstacle.

Critère	longueurs de câbles	distances parcourues
Section flexible 1	(252.57, 290.80, 355.30)	(300, 300, 300)
Section flexible 2	(286.79, 261.00, 200.79)	(231.32, 249.10, 269.30)
Section flexible3	(199.48, 200.36, 200.14)	(178.25, 198.96, 222.15)

3.6.3. Contributions récentes sur la résolution de redondance du modèle géométrique inverse

Le Tableau 3.4 liste quelques contributions traitant le problème de la redondance des robots continuum.

Tableau 3.4 : Quelques contributions traitant le problème de la redondance des robots continuum.

Robot	Approche/méthode	Problème de redondance
Robot continuum actionné par câbles (CDCR) : forme continue, structure rigide, contrôlée par câbles.	Algorithme proposé (sous-algorithme 2)	Peut gérer la redondance en utilisant différents critères.
	Approche basée sur PSO [83].	Peut gérer la redondance en utilisant différents critères.
Compact Bionic Handling Arm (CBHA): forme continue, structure souple, actionnement pneumatique.	Approche basée sur les courbes de l'hodographe de Pythagore [73].	Impossible de gérer le problème de redondance.
	Approche basée sur les réseaux de neurones [105].	
	Approche hybride : Géométrie et réseau de neurones [82].	
	Approche basée sur SQP [84].	

	Approche basée sur PSO [83].	
	Approche basé sur la matrice Jacobéenne [86]	
Bionic Handling Arm (BHA): forme continue, structure souple, actionnement pneumatique.	Goal babbling learning [85].	Peut gérer la redondance en utilisant différents critères.
Manipulateur Redondant : forme continue, structure rigide, multi-sections.	Approche Modale [71].	Impossible de gérer le problème de redondance.
Bras continuum : multi-sections, forme continue, structure souple, trois sections, actionnement pneumatique.	Approche Modale [76].	Impossible de gérer le problème de redondance.
	Approche basée sur la géométrie [75].	

3.7. Conclusion

Dans ce chapitre, nous avons présenté la deuxième partie (sous-algorithme 2) de l'algorithme proposé pour résoudre le problème du modèle géométrique inverse et la redondance d'un robot continuum. Deux critères nommément : minimisation du vecteur des variables des actionneurs et les distances parcourues par les extrémités des premières ($n - 1$) sections flexibles ont été accordés pour résoudre le problème de redondance. En plus, la stratégie d'évitement d'obstacle a été discutée. Pour valider l'efficacité de l'algorithme proposé, des simulations numériques sur des robots à deux et trois sections flexibles sont effectuées dans un environnement libre et contraint.

Chapitre 4

Commande : PID, OPID, FOPID et NMPC

Sommaire

4.1.	Introduction	73
4.2.	Contrôleur PID	73
4.3.	Contrôleur OPID	75
4.4.	Contrôleur PID d'ordre fractionnaire.....	76
4.4.1.	Préliminaires.....	76
4.4.1.1.	L'intégrale fractionnaire de Riemann-Liouville	76
4.4.1.2.	La dérivée fractionnaire de Riemann-Liouville	77
4.4.1.3.	La dérivée fractionnaire de Grünwald-Letnikov	78
4.4.1.4.	La dérivée fractionnaire de Caputo	78
4.4.2.	Structure du contrôleur PID d'ordre fractionnaire	79
4.5.	La commande prédictive	80
4.5.1.	Formulation générale de la commande prédictive non linéaire	81
4.6.	Modèles mathématiques du robot continuum	82
4.6.1.	Positions et orientations	82
4.6.2.	Modèle cinématique	83
4.3.3.	Modèle dynamique	84
4.7.	Applications : implémentation des contrôleurs	85
4.7.1.	Application 1 : implémentation de PID, OPID et FOPID	86
4.7.1.1.	Exemple 1 : suivi d'un point de consigne	87
4.7.1.2.	Exemple 2 : suivi d'une trajectoire	89

4.7.2.	Application 2 : implémentation de NMPC	91
4.7.2.1.	Exemple 1 : suivi d'un point de consigne	92
4.7.2.2.	Exemple 2 : suivi d'une trajectoire	93
4.7.3.	Comparaison des stratégies de commande	96
4.8.	Conclusion	97

4.1. Introduction

Le problème de la commande d'un tel système robotisé peut être formulé comme la détermination de l'évolution des forces/moments généralisées que les actionneurs doivent exercer pour s'assurer que la tâche requise est effectuée de manière précise, fiable et efficace. Cependant, la structure mécanique du robot ainsi que les modèles mathématiques qui régissent son comportement influencent le choix d'un tel contrôleur. Dans ce chapitre, nous allons nous intéresser à l'implémentation en simulation les schémas de commande nommément : Proportionnel-Intégral-Dérivé (PID), PID optimisé (OPID), PID d'Ordre Fractionnaire (FOPID) et la commande prédictive non linéaires (NMPC), afin d'assurer le suivi de trajectoire, via la technique point-à-point, à la fois pour le modèle cinématique et dynamique d'un robot continuum à deux sections. Dans cette étude, la conception des contrôleurs OPID et FOPID est basée sur l'algorithme d'optimisation PSO pour déterminer leurs paramètres optimaux, ainsi que la solution du problème du NMPC.

Dans la première partie de ce chapitre, nous introduisons le principe de base de chaque schéma de commande. Ensuite, les modèles cinématiques et dynamiques du robot étudié sont présentés. Dans la dernière partie du chapitre, nous présentons les résultats de simulation utilisant l'environnement Matlab.

4.2. Contrôleur PID

Le contrôleur Proportionnel-Intégral-Dérivé (PID), appelé aussi correcteur PID, est un organe de contrôle permettant d'effectuer une régulation en boucle fermée en fonction de l'erreur entre la consigne et la sortie du système. La popularité des contrôleurs PID peut être attribuée en partie à leurs performances robustes dans une large gamme de conditions de fonctionnement et en partie à leur simplicité fonctionnelle, qui permet aux ingénieurs de les faire fonctionner de manière simple et directe.

Comme son nom l'indique, le contrôleur PID se compose de trois paramètres : Proportionnel (P), Intégral (I) et Dérivé (D) dont chacun a son propre rôle. Par exemple, le paramètre (P) dépend de l'erreur actuelle et il s'agit sur la rapidité du système, c.-à-d. il diminue de façon importante le temps de réponse du système. Le paramètre (I) dépend de l'accumulation des erreurs passées et son rôle est d'améliorer l'erreur en régime permanent, et le paramètre (D) est une prévision d'erreurs future et qui s'assure l'amélioration de la stabilité du système. Le processus de réglage des gains optimaux pour les paramètres P, I et D pour

obtenir une réponse optimale d'un système est appelé réglage. Il existe différentes méthodes de réglage dont la méthode la plus utilisée est la méthode de Ziegler Nichols.

En général, le comportement d'un système à commander est évalué selon trois critères qui sont :

- **La stabilité** : un système est stable lorsque sa réponse est converge à son état d'équilibre pour une consigne constante.
- **La précision** : un système est précis si la sortie suit la consigne avec un écart inférieur à une valeur prédéfinie.
- **La rapidité** : la rapidité signifie le temps de réaction de la sortie du système par rapport à la consigne. Plus précisément, elle s'exprime par le temps de réponse qui est le temps mise par le système pour ne plus dépasser à 5 % de la valeur finale.

Le Tableau 4.1 montre l'influence des trois paramètres du contrôleur PID sur le réglage de la sortie d'untel système.

Tableau 4.1 : Influence des paramètres du contrôleur PID.

	Stabilité	Précision	Rapidité
K_p	Diminue	Augmente	Augmente
K_i	Diminue	Augmente	Diminue
K_d	Augmente	Pas d'influence	Augmente

La loi de commande du contrôleur PID dans le domaine temporel est donnée par l'équation suivante :

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (4.1)$$

telles que :

- $u(t)$: représente la commande générée par le contrôleur ;
- $e(t)$: est l'erreur à l'instant t ;
- K_p, K_i, K_d : représentent respectivement les paramètres Proportionnel, Intégral et Dérivé du contrôleur.

La combinaison de ces trois coefficients constitue le mécanisme d'ajustement des performances du système comme montre la Figure 4.1.

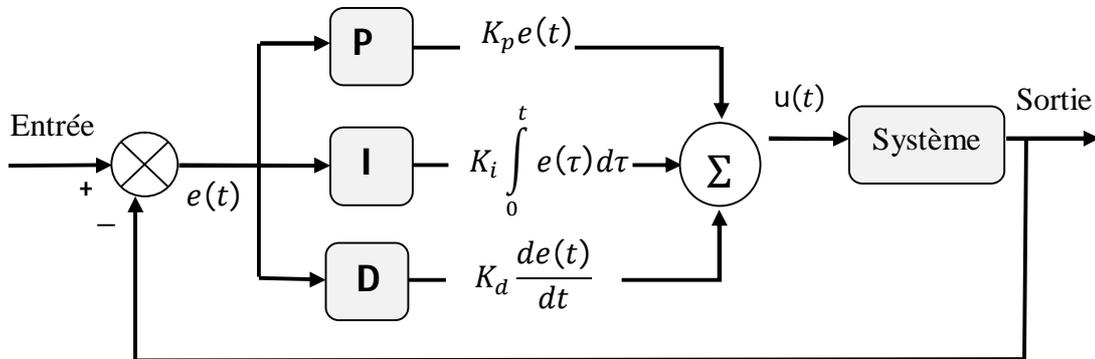


Figure 4.1 : Structure du contrôleur PID.

4.3. Contrôleur OPID

Afin d'améliorer les performances du contrôleur PID classique, les trois paramètres P, I et D sont déterminés par la méthode PSO, d'où le nom contrôleur PID optimisé (OPID). La structure de réglage de ces trois paramètres du contrôleur PID à l'aide de la méthode PSO en boucle fermée est illustrée schématiquement sur la Figure 4.2.

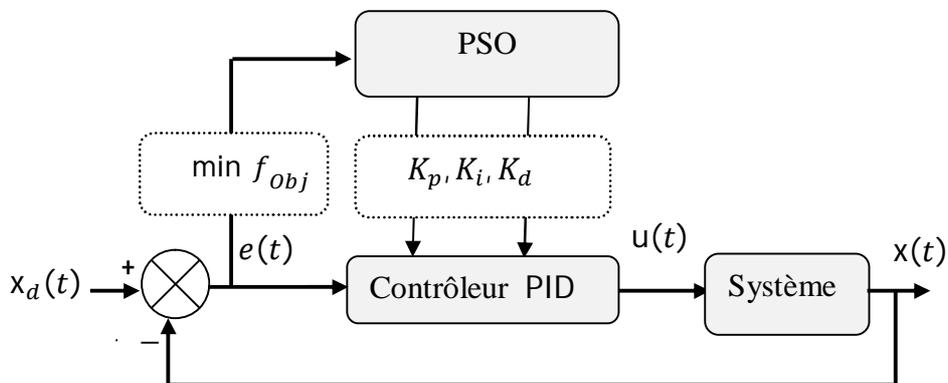


Figure 4.2 : Structure de réglage des paramètres du contrôleur PID.

En pratique, l'étape principale de l'application de la méthode PSO est de choisir la meilleure fonction objective pour évaluer la fitness de chaque particule. Plusieurs critères peuvent être utilisés à savoir :

– L'intégrale de la valeur absolue de l'erreur, en Anglais "Integral of Absolute Error (IAE)" :

$$IAE = \int_0^{\infty} |e(t)| dt \quad (4.2)$$

- L'intégrale du carré de l'erreur, en Anglais "Integral of Squared Error (ISE)" :

$$ISE = \int_0^{\infty} e(t)^2 dt \quad (4.3)$$

- L'intégrale de la valeur absolue de l'erreur pondérée par le temps, en Anglais "Integral of Time multiplied by Absolute Error (ITAE)" :

$$ITAE = \int_0^{\infty} t. |e(t)| dt \quad (4.4)$$

- L'intégrale du carré de l'erreur pondérée par le temps, en Anglais "Integral of Time Squared Error (ITSE)" :

$$ITSE = \int_0^{\infty} t. e(t)^2 dt \quad (4.5)$$

tel que : $e(t)$ est l'erreur entre la consigne et la sortie du système commandé, et t est le temps.

4.4. Contrôleur PID d'ordre fractionnaire

4.4.1. Préliminaires

Contrairement aux définitions traditionnelles des opérateurs de calcul intégral et dérivé, le calcul fractionnaire est un concept mathématique traitant des intégrales et des dérivées d'ordres non entier ${}_{t_0}D_t^m$, où t_0 et t sont des limites de l'opération.

Dans le cas général, l'opérateur intégral-différentiel continu est défini par [20], [21] :

$${}_{t_0}D_t^\alpha = \begin{cases} \frac{d^\alpha}{dt^\alpha} & R(\alpha) > 0 \\ 1 & R(\alpha) = 0 \\ \int_{t_0}^t (d\tau)^{-\alpha} & R(\alpha) < 0 \end{cases} \quad (4.6)$$

où α est un nombre complexe représente l'ordre de l'opération, et $R(m)$ est la partie réelle d'un nombre complexe.

Il existe plusieurs définitions mathématiques pour l'intégration et dérivation d'ordre fractionnaire, dont les plus célèbres sont :

4.4.1.1. L'intégrale fractionnaire de Riemann-Liouville

Les intégrales fractionnaires, à droite et à gauche, de Riemann-Liouville de la fonction $f: [a, b] \rightarrow \mathbb{R}$ sont exprimées en fonction de l'ordre de dérivée α comme suivant :

Intégrale à droite :

$${}^{RL}D_t^{-\alpha} f(t) = \frac{1}{\Gamma(\alpha)} \int_a^t (t-s)^{\alpha-1} f(s) ds, \quad t > a \quad (4.7)$$

Intégrale à gauche :

$${}^{RL}D_b^{-\alpha} f(t) = \frac{1}{\Gamma(\alpha)} \int_t^b (s-t)^{\alpha-1} f(s) ds, \quad t < b \quad (4.8)$$

avec $\Gamma(\alpha)$ est la fonction Gamma donné par :

$$\Gamma(\alpha) = \int_0^{+\infty} e^{-t} t^{\alpha-1} dt \quad (4.9)$$

4.4.1.2. La dérivée fractionnaire de Riemann-Liouville

Les dérivées fractionnaires, à droite et à gauche, de Riemann-Liouville de la fonction $f: [a, b] \rightarrow \mathbb{R}$ sont exprimées en fonction de l'ordre de dérivée positif α comme suivant :

1^{er} Cas : $\alpha \geq 1$

Dérivée à droite :

$${}^{RL}D_t^\alpha f(t) = \frac{d^n}{dt^n} {}^{RL}D_t^{-(n-\alpha)} f(t) \quad (4.10)$$

ou bien :

$${}^{RL}D_t^\alpha f(t) = \frac{1}{\Gamma(n-\alpha)} \frac{d^n}{dt^n} \int_a^t (t-s)^{n-\alpha-1} f(s) ds, \quad t > a \quad (4.11)$$

Dérivée à gauche :

$${}^{RL}D_b^\alpha f(t) = (-1)^n \frac{d^n}{dt^n} {}^{RL}D_b^\alpha f(t) \quad (4.12)$$

ou bien :

$${}^{RL}D_b^\alpha f(t) = \frac{1}{\Gamma(n-\alpha)} (-1)^n \frac{d^n}{dt^n} \int_t^b (s-t)^{n-\alpha-1} f(s) ds, \quad t < b \quad (4.13)$$

tel que : $n = [\alpha] + 1$, où $[\alpha]$ est la partie entière de α .

2^{ème} Cas : $0 < \alpha < 1$

Dérivée à droite :

$${}^{RL}D_t^\alpha f(t) = \frac{1}{\Gamma(1-\alpha)} \frac{d}{dt} \int_a^t (t-s)^{-\alpha} f(s) ds, \quad t > a \quad (4.14)$$

Dérivée à gauche :

$${}^{RL}D_b^\alpha f(t) = -\frac{1}{\Gamma(1-\alpha)} \frac{d}{dt} \int_t^b (s-t)^{-\alpha} f(s) ds, \quad t < b \quad (4.15)$$

4.4.1.3. La dérivée fractionnaire de Grünwald-Letnikov

Les dérivées fractionnaires, à droite et à gauche, de Grünwald-Letnikov de la fonction $f: [a, b] \rightarrow \mathbb{R}$ sont exprimées en fonction de l'ordre de dérivée positif α comme suivant :

Dérivée à droite :

$${}^{GL}D_t^\alpha f(t) = \lim_{h \rightarrow 0} \frac{1}{h^\alpha} \sum_{k=0}^{(t-a)/h} (-1)^k \binom{\alpha}{k} f(t - kh), \quad t > a \quad (4.16)$$

Dérivée gauche :

$${}^{GL}D_b^\alpha f(t) = \lim_{h \rightarrow 0} \frac{1}{h^\alpha} \sum_{k=0}^{(b-t)/h} (-1)^k \binom{\alpha}{k} f(t - kh), \quad t < b \quad (4.17)$$

4.4.1.4. La dérivée fractionnaire de Caputo

Les dérivées fractionnaires, à droite et à gauche, de Caputo de la fonction $f: [a, b] \rightarrow \mathbb{R}$ sont exprimées en fonction de l'ordre de dérivée positif α comme suivant :

1^{er} Cas : $\alpha \geq 1$:

Dérivée à droite :

$${}^C D_t^\alpha f(t) = -\frac{1}{\Gamma(n-\alpha)} \int_a^t f^{(n)}(s) (t-s)^{n-\alpha-1} ds, \quad t > a \quad (4.18)$$

Dérivée à gauche :

$${}_t^c D_b^\alpha f(t) = -\frac{1}{\Gamma(n-\alpha)} \int_t^b f^{(n)}(s)(s-t)^{n-\alpha-1} ds, \quad t < b \quad (4.19)$$

2^{ier} Cas : $0 < \alpha < 1$

Dérivée à droite :

$${}_a^c D_t^\alpha f(t) = -\frac{1}{\Gamma(1-\alpha)} \int_a^t f'(s)(t-s)^{-\alpha} ds \quad (4.20)$$

Dérivée à gauche :

$${}_t^c D_b^\alpha f(t) = -\frac{1}{\Gamma(1-\alpha)} \int_t^b f'(s)(s-t)^{-\alpha} ds \quad (4.21)$$

4.4.2. Structure du contrôleur PID d'ordre fractionnaire

Le contrôleur PID d'ordre fractionnaire, ou en anglais Fractional Order PID (FOPID), est symbolisé par $PI^\lambda D^\mu$. Il s'agit d'une généralisation du contrôleur PID classique dans laquelle l'intégrale et la dérivée ordinaire sont remplacés par des opérateurs fractionnaires λ et μ respectivement. Ces deux paramètres peuvent être donnés avec n'importe quel nombre réel arbitraire où leurs valeurs sont choisies généralement dans la plage 0 à 2. L'ajout de ces deux paramètres offre au contrôleur une grande flexibilité et conduit à des bons résultats et plus grande robustesse pour le système contrôlé. La Figure 4.3 présente la relation entre les contrôleurs PID et FOPID en fonction des deux paramètres λ et μ .

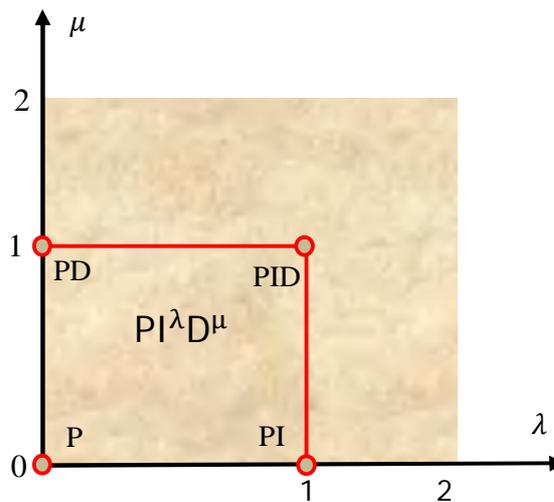


Figure 4.3 : Relation entre les contrôleurs PID et FOPID.

D'après cette Figure 4.3, cinq cas peuvent être distingués en fonction des deux opérateurs fractionnaires λ et μ . Lorsque les valeurs des deux paramètres (λ, μ) sont égales à 1, le contrôleur FOPID devient le PID classique. Les trois classes du contrôleur PID classique, nommément : le contrôleur proportionnels (P), le contrôleur proportionnel-intégral (PI) et le contrôleur proportionnel-dérivé (PD), sont obtenues lorsque les valeurs des deux paramètres (λ, μ) sont égales à (0, 0), (1, 0) et (0, 1), respectivement.

La loi de commande du contrôleur FOPID dans le domaine temporel est donnée par l'équation suivante :

$$x(t) = K_p e(t) + K_i D^{-\lambda} e(t) + K_d D^\mu e(t) \quad (4.22)$$

où K_p , K_i et K_d représentent les gains proportionnel, intégration et différentiation, respectivement. λ et μ sont les ordres fractionnaires de l'action de l'intégration et la différentiation, respectivement.

Cependant, malgré l'efficacité du contrôleur FOPID par rapport au PID classique, le réglage de ces paramètres reste une tâche difficile. Dans ce travail, la méthode PSO est utilisée pour ajuster les paramètres du contrôleur FOPID due à sa simplicité. La structure générale de réglage des paramètres du contrôleur FOPID par l'algorithme PSO en boucle fermée est présentée sur la Figure 4.4.

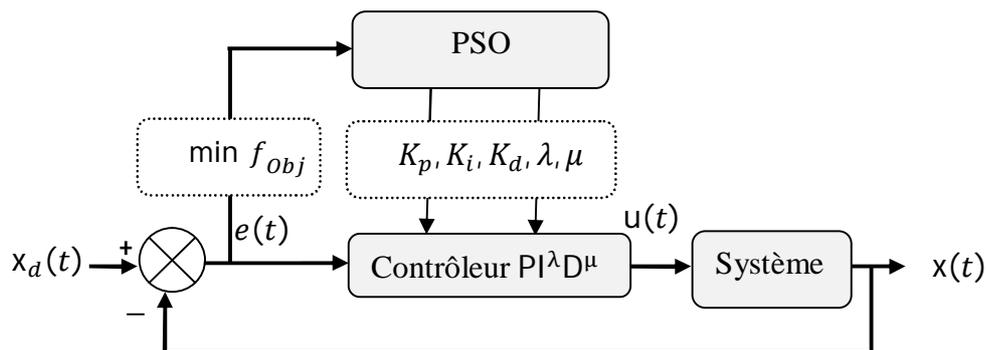


Figure 4.4 : Structure de réglage des paramètres du contrôleur FOPID par la méthode PSO.

4.5. La commande prédictive

Le concept de base de la commande prédictive est d'utiliser un modèle mathématique, linéaire ou non linéaire, pour prévoir le comportement du système afin de sélectionner les entrées de contrôleur en minimisant une fonction objective. Généralement, la fonction

objective est donnée par l'écart entre la prédiction générée et un point de la trajectoire de référence à suivre, tout en respectant des contraintes de fonctionnement. Comme l'état du système est mis à jour à chaque période d'échantillonnage, le problème d'optimisation doit être résolu à chaque intervalle d'échantillonnage. Dans cette section, on s'intéresse uniquement à la commande prédictive non linéaire, ou en Anglais Nonlinear Model Predictive Control (NMPC).

4.5.1. Formulation générale de la commande prédictive non linéaire

Le modèle de mouvement discret d'un tel système non linéaire peut être décrit comme suit :

$$x_{k+1} = f(x_k, u_k) \tag{4.23}$$

où $x_k \in X$ est l'état du système, $u_k \in U$ est le signal de commande, et f est une fonction continue. X et U représentent respectivement l'ensemble des valeurs d'état et d'entrée faisables au système.

Le problème d'optimisation à résoudre par le NMPC peut être formulé comme suivant :

$$\min_U f_{Obj}(x_k, u_k, k) \tag{4.24}$$

tel que : f_{Obj} est donnée par :

$$f_{Obj}(x_k, u_k, k) = W(x_{k+N}) + \sum_{s=k}^{k+N-1} F_N(x_s, u_s) \tag{4.25}$$

où f_{Obj} est la fonction objective à minimisée, N est l'horizon de prédiction, et $W(x_{k+N})$ est un poids sur l'état final.

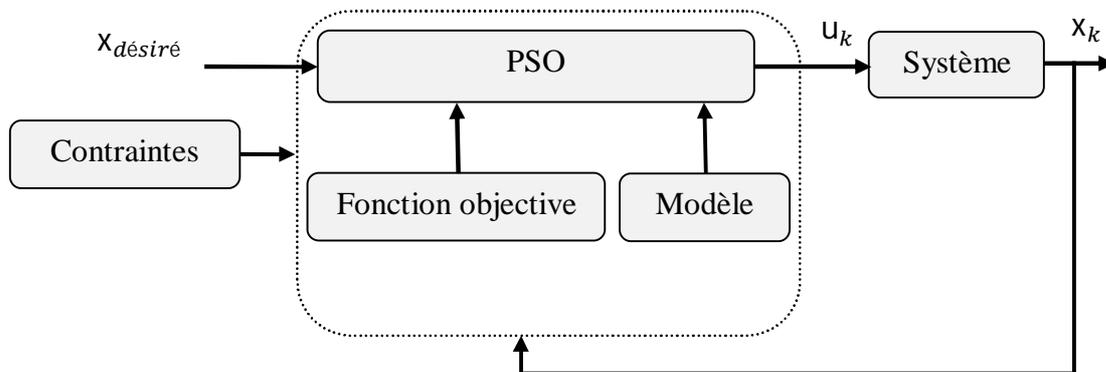


Figure 4.5 : Structure de la commande NMPC avec la méthode PSO.

La solution du problème d'optimisation décrit dans l'équation (4.25) permet un suivi précis de la trajectoire de référence désirée, et permet de calculer le signal de commande à travers l'horizon de prédiction. Pour chaque échantillonnage, le problème d'optimisation est résolu pour obtenir des séquences de contrôle optimales, mais seule la première séquence est appliquée au système. Le schéma fonctionnel du NMPC basé sur la méthode PSO est présenté sur la Figure. 4.5.

4.6. Modèles mathématiques du robot continuum

Le principe général de n'importe quel contrôleur est d'utiliser un modèle mathématique pour produire la meilleure décision à l'heure actuelle. Les modèles sont donc au cœur de toute forme de contrôleur. Dans cette section, nous introduisons brièvement le modèle géométrique, cinématique et dynamique du robot continuum actionné par câbles à deux sections flexibles afin de mettre en œuvre les contrôleurs proposés dans les sections précédentes.

4.6.1. Positions et orientations

Pour calculer le modèle cinématique du robot, il faut d'abord connaître les coordonnées locales et globales de chaque section flexible k . En se référant aux chapitres précédents, les coordonnées locales du vecteur de position $\mathbf{P}_{k,l}$ de l'extrémité libre de chaque section flexible k , associés au repère de référence \mathcal{R}_k par rapport au repère \mathcal{R}_{k-1} , peuvent être exprimés par l'Equation suivantes :

$$\mathbf{P}_{k,l} = \begin{cases} X = \frac{\ell_k}{\theta_k} (1 - \cos(\theta_k)) \cos(\varphi_k) \\ Y = \frac{\ell_k}{\theta_k} (1 - \cos(\theta_k)) \sin(\varphi_k) \\ Z = \frac{\ell_k}{\theta_k} \sin(\theta_k) \end{cases} \quad (4.26)$$

et la matrice de rotation $\mathbf{R}_{k,l}$ peut être donnée par la multiplication successive des trois matrices de rotation suivantes :

$$\begin{aligned} \mathbf{R}_{k,l} &= \mathbf{rot}(Z_{k-1}, \varphi_k) \mathbf{rot}(Y_{k-1}, \theta_k(s)) \mathbf{rot}(Z_{k-1}, -\varphi_k) \\ &= \begin{bmatrix} C\varphi_k & -S\varphi_k & 0 \\ S\varphi_k & C\varphi_k & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C\theta_k & 0 & S\theta_k \\ 0 & 1 & 0 \\ -S\theta_k & 0 & C\theta_k \end{bmatrix} \begin{bmatrix} C\varphi_k & S\varphi_k & 0 \\ -S\varphi_k & C\varphi_k & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned} \quad (4.27)$$

Plus généralement, la matrice $\mathbf{R}_{k,l}$ peut être écrite en fonction des vecteurs unitaires $\mathbf{n}_{k,l}$, $\mathbf{b}_{k,l}$ et $\mathbf{t}_{k,l}$ comme montre l'Equation suivante.

$$\mathbf{R}_{k,l} = [\mathbf{n}_{k,l} \quad \mathbf{b}_{k,l} \quad \mathbf{t}_{k,l}] \quad (4.28)$$

tels que : $\mathbf{n}_{k,l}$, $\mathbf{b}_{k,l}$ et $\mathbf{t}_{k,l}$ sont le vecteur normal, le vecteur bi-normal et le vecteur tangent représentant respectivement la 1^{ière}, 2^{ième} et 3^{ième} colonne de la matrice $\mathbf{R}_{k,l}$.

Les coordonnées globales consiste à calculer la position et l'orientation de l'extrémité libre de chaque section flexible k du robot par rapport au repère de référence \mathcal{R}_0 peuvent être exprimées comme suivant.

$$\mathbf{P}_k = \begin{cases} \mathbf{P}_{1,l} & k = 1 \\ \mathbf{P}_{1,l} + \mathbf{R}_1 \mathbf{P}_{2,l} & k = 2 \end{cases} \quad (4.29)$$

$$\mathbf{R}_k = \begin{cases} \mathbf{R}_{1,l} & k = 1 \\ \mathbf{R}_1 \mathbf{R}_{2,l} & k = 2 \end{cases} \quad (4.30)$$

4.6.2. Modèle cinématique

Littéralement, le modèle cinématique est un modèle des vitesses et mathématiquement est une différentiation du modèle géométrique du robot c.à.d. les coordonnées locales et globales. Ainsi, la vitesse linéaire $\mathbf{v}_{k,l}$ par rapport à l'origine du repère local \mathcal{R}_{k-1} peut être calculée par la différentiation par rapport au temps de l'équation (4.26), ce qui donne :

$$\mathbf{v}_{k,l} = \begin{cases} \dot{X} = \dot{\theta}_k \left[\frac{\ell_k}{\theta_k} \sin(\theta_k) - \frac{\ell_k}{\theta_k^2} (1 - \cos(\theta_k)) \right] \cos(\varphi_k) \\ \quad - \frac{\ell_k}{\theta} (1 - \cos(\theta_k)) \dot{\varphi} \sin(\varphi) \\ \dot{Y} = \dot{\theta}_k \left[\frac{\ell_k}{\theta_k} \sin(\theta_k) - \frac{\ell_k}{\theta_k^2} (1 - \cos(\theta_k)) \right] \sin(\varphi_k) \\ \quad + \frac{\ell_k}{\theta} (1 - \cos(\theta_k)) \dot{\varphi} \cos(\varphi) \\ \dot{Z} = \dot{\theta}_k \left[\frac{\ell_k}{\theta_k} \cos(\theta_k) - \frac{\ell_k}{\theta_k^2} \sin(\theta_k) \right] \end{cases} \quad (4.31)$$

et la vitesse angulaire $\boldsymbol{\omega}_{k,l}$ peut être calculée comme suivant :

$$\boldsymbol{\omega}_{k,l} = \hat{\mathbf{t}}_{k,l} \dot{\mathbf{t}}_{k,l} \quad (4.32)$$

tel que $\dot{\mathbf{t}}_{k,l}$ est la dérivée par rapport au temps du vecteur $\mathbf{t}_{k,l}$ et $\hat{\mathbf{t}}_{k,l}$ est la matrice antisymétrique associée au vecteur $\mathbf{t}_{k,l}$. Cette matrice est donnée par l'équation suivante :

$$\hat{\mathbf{t}}_{k,l} = \begin{bmatrix} 0 & -\cos(\theta_k) & \sin(\varphi_k)\sin(\theta_k) \\ \cos(\theta_k) & 0 & -\cos(\varphi_k)\sin(\theta_k) \\ -\sin(\varphi_k)\sin(\theta_k) & \cos(\varphi_k)\sin(\theta_k) & 0 \end{bmatrix} \quad (4.33)$$

De même manière, les vitesses linéaires et angulaires par rapport au repère de référence \mathcal{R}_0 peuvent être calculées récursivement par la différenciation par rapport au temps comme suivant :

$$\mathbf{v}_k = \begin{cases} \mathbf{v}_1 & k = 1 \\ \mathbf{v}_1 + \hat{\boldsymbol{\omega}}_1 \mathbf{R}_1 \mathbf{P}_{2,l} + \mathbf{R}_1 \dot{\mathbf{v}}_{2,l} & k = 2 \end{cases} \quad (4.34)$$

$$\boldsymbol{\omega}_k = \begin{cases} \boldsymbol{\omega}_1 & k = 1 \\ \boldsymbol{\omega}_1 + \mathbf{R}_1 \boldsymbol{\omega}_{2,l} & k > 1 \end{cases} \quad (4.35)$$

tel que :

$$\hat{\boldsymbol{\omega}}_1 = \begin{bmatrix} 0 & -\boldsymbol{\omega}_z & \boldsymbol{\omega}_y \\ \boldsymbol{\omega}_z & 0 & -\boldsymbol{\omega}_x \\ -\boldsymbol{\omega}_y & \boldsymbol{\omega}_x & 0 \end{bmatrix} \quad (4.36)$$

4.6.3. Modèle dynamique

Dans cette section, le modèle dynamique d'un robot continuum planaire à deux sections flexibles est présenté. Ainsi, pour les coordonnées généralisées $q = [\theta_1, \theta_2]$, les équations du mouvement régissant le comportement du robot selon la méthode Lagrange-Euler sont données par le système d'équations suivant [55]:

$$\begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{Bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{Bmatrix} + \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \end{bmatrix} \begin{Bmatrix} \dot{\theta}_1^2 \\ \dot{\theta}_1 \dot{\theta}_2 \\ \dot{\theta}_2^2 \end{Bmatrix} + \begin{bmatrix} K_1 \\ K_2 \end{bmatrix} = \begin{Bmatrix} Q_1 \\ Q_2 \end{Bmatrix} \quad (4.37)$$

où les éléments intervenant dans les matrices décrites dans l'équation (4.37) sont donnés dans l'annexe.

Pour résoudre ce système d'équations, la méthode de Runge-Kutta d'ordre 4 peut être utilisée comme solution numérique. Pour cela, les variables d'état sont données comme suivant:

$$\begin{cases} u_1(t) = \theta_1(t), & u_3(t) = \theta_2(t) \\ u_2(t) = \dot{\theta}_1(t), & u_4(t) = \dot{\theta}_2(t) \end{cases} \quad (4.38)$$

En introduisant ces variables d'état dans le système d'équations (4.37), on obtient :

$$\begin{cases} \dot{u}_1(t) = u_2 \\ \dot{u}_2(t) = \frac{1}{M_{11}} \left(Q_1 - K_1 - C_{13}u_4^2 - C_{12}u_2u_4 - C_{11}u_2^2 - \frac{M_{12}}{(M_{12}M_{21} - M_{11}M_{22})} (Q_1M_{21} \right. \\ \quad \left. + K_2M_{11} - Q_2M_{11} - K_1M_{21} - (C_{13}M_{21} - C_{23}M_{11})u_4^2 \right. \\ \quad \left. - (C_{11}M_{21} - C_{21}M_{11})u_2^2 - C_{13}M_{21}u_2u_4 \right) \\ \dot{u}_3(t) = u_4 \\ \dot{u}_4(t) = \frac{1}{(M_{12}M_{21} - M_{11}M_{22})} \left(Q_1M_{21} + K_2M_{11} - Q_2M_{11} - K_1M_{21} - (C_{13}M_{21} \right. \\ \quad \left. - C_{23}M_{11})u_4^2 - (C_{11}M_{21} - C_{21}M_{11})u_2^2 - C_{13}M_{21}u_2u_4 \right) \end{cases} \quad (4.39)$$

ou sous forme compact, l'équation (4.39) s'écrit :

$$\dot{u}(t) = H(U, t) + G(u, t)Q(t) \quad (4.40)$$

tels que :

$$\dot{u}(t) = [\dot{u}_1 \quad \dot{u}_2 \quad \dot{u}_3 \quad \dot{u}_4]^T \quad (4.41)$$

$$H(u, t) = \begin{cases} u_2 \\ \frac{1}{M_{11}} \left(-K_1 - C_{13}u_4^2 - C_{12}u_2u_4 - C_{11}u_2^2 - \frac{M_{12}}{(M_{12}M_{21} - M_{11}M_{22})} (K_2M_{11} - K_1M_{21} \right. \\ \quad \left. - (C_{13}M_{21} - C_{23}M_{11})u_4^2 - (C_{11}M_{21} - C_{21}M_{11})u_2^2 - C_{13}M_{21}u_2u_4 \right) \\ u_4 \\ \frac{1}{(M_{12}M_{21} - M_{11}M_{22})} \left(K_2M_{11} - K_1M_{21} - (C_{13}M_{21} - C_{23}M_{11})u_4^2 - C_{13}M_{21}u_2u_4 \right. \\ \quad \left. - (C_{11}M_{21} - C_{21}M_{11})u_2^2 \right) \end{cases} \quad (4.42)$$

$$G(u, t) = \begin{bmatrix} 0 & 0 & 0 \\ \frac{1}{M_{11}} \left(1 - \frac{M_{12}M_{21}}{(M_{12}M_{21} - M_{11}M_{22})} \right) & 0 & -\frac{M_{11}M_{12}}{(M_{12}M_{21} - M_{11}M_{22})} & 0 \\ 0 & 0 & 0 & 0 \\ \frac{0}{(M_{12}M_{21} - M_{11}M_{22})} & 0 & \frac{-M_{11}}{(M_{12}M_{21} - M_{11}M_{22})} & 0 \end{bmatrix} \quad (4.43)$$

$$Q(t) = \begin{cases} Q_1 \\ 0 \\ Q_2 \\ 0 \end{cases} \quad (4.44)$$

où $u(t)$ est le vecteur de variable d'état, $H(u, t)$ et $G(u, t)$ sont des fonctions non linéaires, $Q(t)$ est le vecteur de commande.

4.7. Applications : implémentation des contrôleurs

Dans cette section, les quatre contrôleurs présentés précédemment nommément : PID, OPID, FOPID et NMPC sont utilisés pour commander un robot continuum à deux sections flexibles. Les caractéristiques géométriques et inertielles du robot sont définies dans le Tableau 4.2. La commande par l'utilisation des contrôleurs PID, OPID et FOPID sont associés au modèle dynamique, alors que le contrôleur NMPC est associé aux deux modèles cinématique et dynamique. Les simulations sont effectuées dans le logiciel MATLAB sur un ordinateur avec les spécifications suivantes : Intel(R) Xeon(R) CPU E5-1620 0 @ 3,60 GHz, 16 Go de RAM, système d'exploitation 64 bits.

Tableau 4.2 : Paramètres du robot continuum [55].

Paramètre	Désignation	Valeur
ℓ_k	Longueur de la section flexible	300 mm
m	Masse du disque	10 grammes
g	Constante de gravité	9.81 m/s ²
E	Module de Young	2.1×10^{11} Pa
I_b	Moment d'inertie du disque	12.56×10^{-12} m ⁴
I_{xx}	Moment d'inertie du disque	3.068×10^{-7} m ⁴
I_t	Moment d'inertie de la tige flexible	$3.97 \cdot 10^{-12}$ m ⁴

Avec : $k = 1,2$

4.7.1. Application 1 : implémentation de PID, OPID et FOPID

Dans cette application, nous comparons les performances des trois contrôleurs, nommément PID, OPID et FOPID pour la commande d'un robot continuum à deux sections flexibles. Pour ce faire, deux exemples sont considérés. Le premier exemple est proposé pour stabiliser le point final du robot à une cible prédéfinie (suivi d'un point de consigne), et le deuxième exemple est considéré pour un suivi d'une trajectoire. Il est à noter que pour chaque type contrôleur, il y a deux sous-contrôleurs. L'un pour la première section flexible du robot et l'autre pour la deuxième section comme montre la Figure 4.6. Les paramètres optimaux des trois contrôleurs sont donnés dans le Tableau 4.3, et la période d'échantillonnage aussi pour les trois contrôleurs est 0.005 sec.

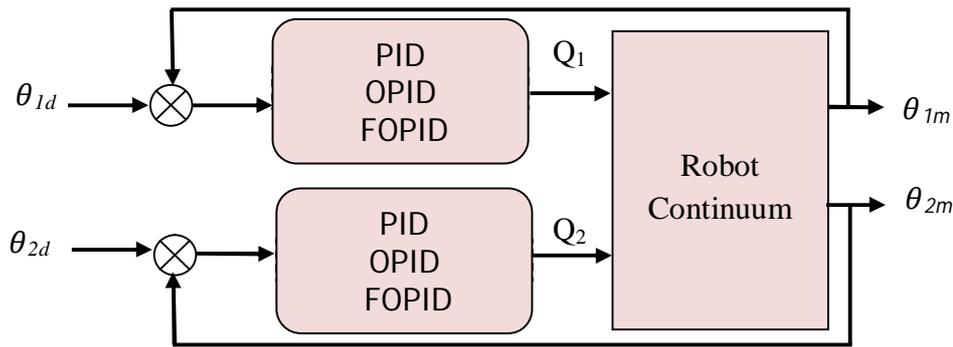


Figure 4.6 : Structure des commandes PID, OPID et FOPID en boucle fermée.

Tableau 4.3 : Les paramètres optimaux des contrôleurs PID, OPID et FOPID.

	PID	OPID	FOPID
λ	1	1	0.861
μ	1	1	0.973
K_p	2.186	0.196	1.486
K_i	1.225	3.705	2.225
K_d	3.745	0.658	2.905

4.7.1.1. Exemple 1 : suivi d'un point de consigne

L'objectif de cet exemple est d'amener la sortie du robot θ_1 et θ_2 assez proche que possible des valeurs de consigne données par $\theta_{1,d} = \theta_{2,d} = \pi/4$, suivi de $\theta_{2,d} = \pi/6$ et $\theta_{2,d} = \pi/3$.

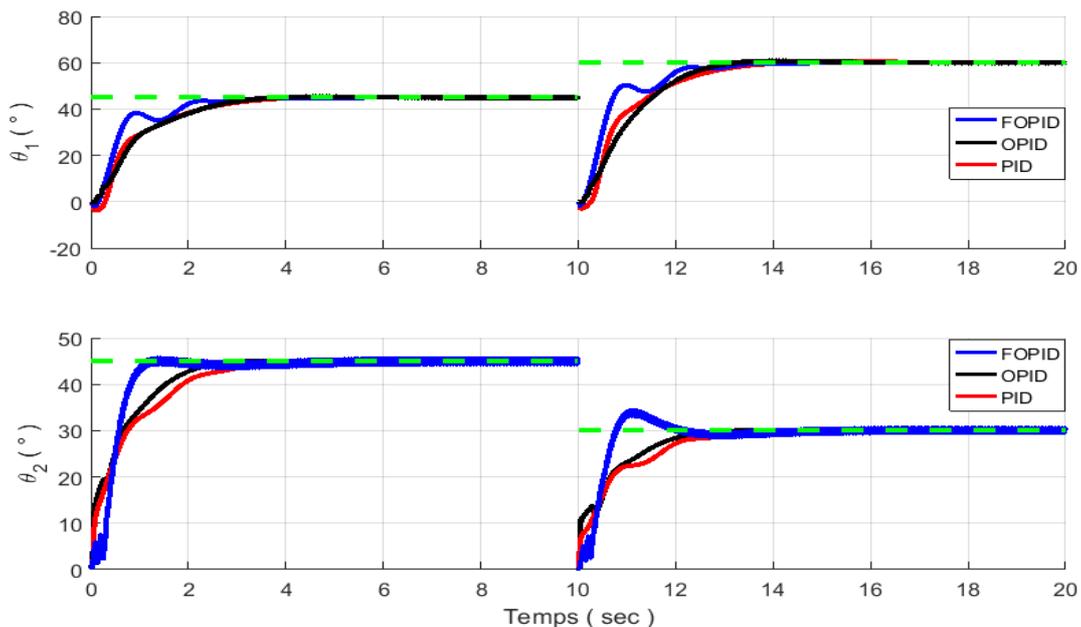


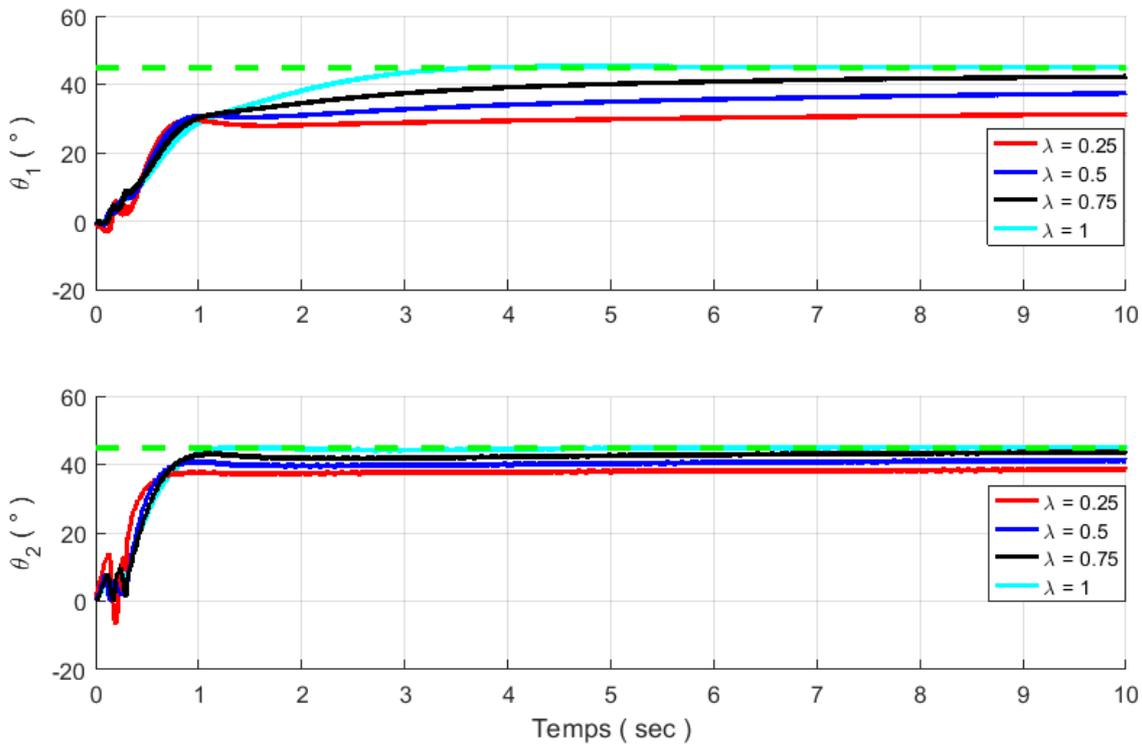
Figure 4.7 : Comparaison des réponses des contrôleurs PID, OPID et FOPID pour un suivi d'un point de consigne.

La Figure 4.7 présente la réponse dynamique en boucle fermée des trois contrôleurs PID, OPID et FOPID, dans lesquels l'erreur quadratique moyenne (MSE) est calculée (Tableau 4.4). D'après les résultats de ce Tableau, nous pouvons constater que la supériorité de performance est obtenue par le contrôleur FOPID puis le contrôleur OPID que PID classique. Ce résultat confirme que le contrôleur FOPID peut être utilisé avec succès pour commander ce type de robots pour des applications de suivi de trajectoires.

Tableau 4.4 : MSE pour un suivi d'un point de consigne.

		contrôleurs		
		PID	OPID	FOPID
Erreur quadratique moyenne (MSE)	θ_1	0.0932	0.0768	0.0665
	θ_2	0.0795	0.0909	0.0722

Pour mieux montrer l'influence des deux paramètres λ et μ sur le contrôleur FOPID, on varie la valeur de l'un de ces paramètres tout en gardant la valeur de l'autre paramètre constante est égale à 1. Les Figures 4.8 et 4.9) montrent l'influence de ces deux paramètres sur la réponse du contrôleur pour un suivi d'un point de consigne.

Figure 4.8 : Réponses du contrôleur FOPID avec $\mu = 1$ et différentes valeurs de λ .

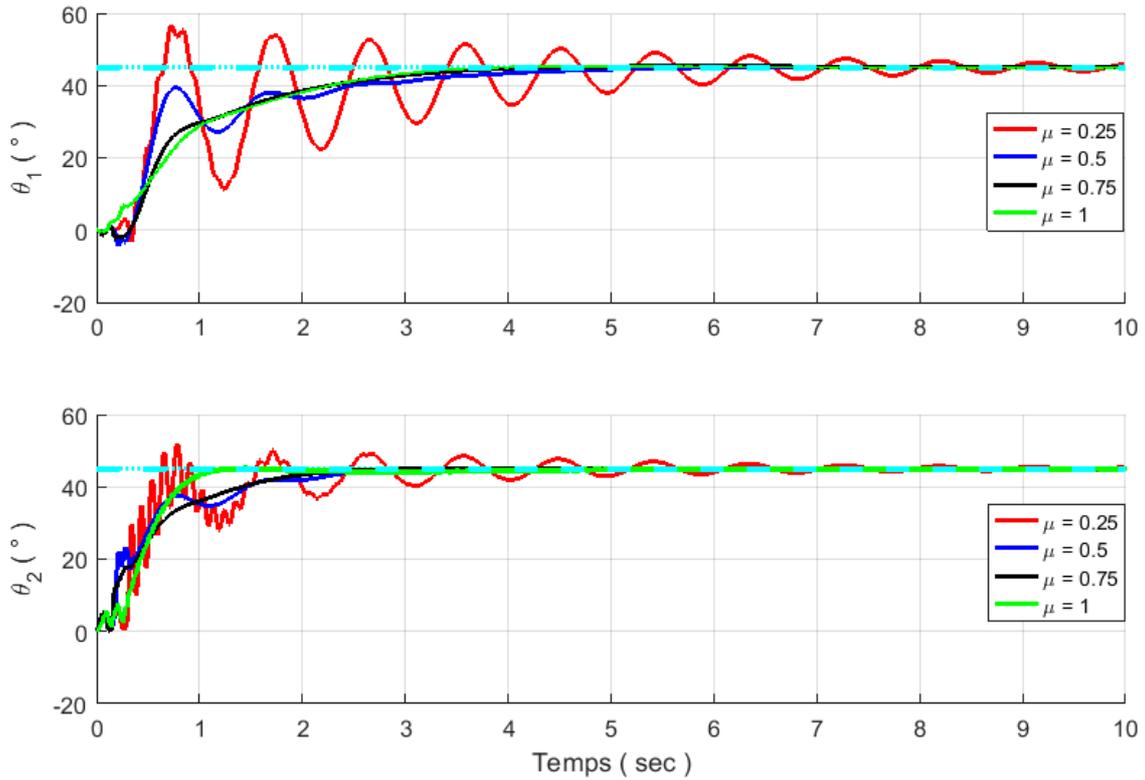


Figure 4.9 : Réponses du contrôleur FOPID avec $\lambda = 1$ et différentes valeurs de μ .

4.7.1.2. Exemple 2 : suivi d'une trajectoire

Pour illustrer davantage l'efficacité du contrôleur FOPID, nous l'utiliserons pour un suivi d'une trajectoire décrite mathématiquement par l'équation (4.45), dont le quel le robot part de la position initiale comme $[0, 0, 600]$ (mm).

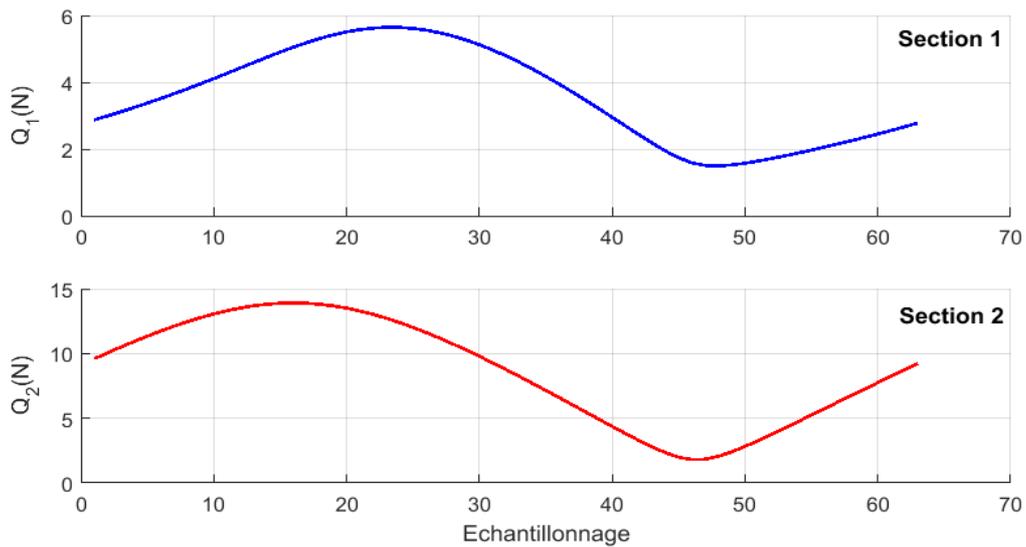


Figure 4.10 : Signal de commande nécessaire pour un suivi d'une trajectoire.

$$x_d = \begin{cases} 100\sin\left(\frac{\pi t}{5}\right) \\ 0 \\ 500 + 100\cos\left(\frac{\pi t}{5}\right) \end{cases} \quad (4.45)$$

Pour montrer la performance du contrôleur FOPID en termes de précision du suivi, la Figure 4.11 présente en évidence la trajectoire désirée et générée, ainsi que les erreurs Euclidiennes entre elles le long des axes x et z . D'après cette Figure, on observe que les courbes sont bien superposées et que les erreurs moyennes sont inférieures à 0,016 mm et 0,0031 mm le long de l'axe des x et de l'axe des z , respectivement. Ce résultat indique la bonne performance du contrôleur FOPID en termes de précision du suivi. En outre, la Figure 4.12 présente davantage les angles de flexion désiré et généré pour les deux sections flexibles du robot, ainsi que les erreurs Euclidiennes entre elles. Il convient de noter que la trajectoire générée est réalisée à l'aide du modèle dynamique basé sur les résultats obtenus à partir de la simulation de commande FOPID.

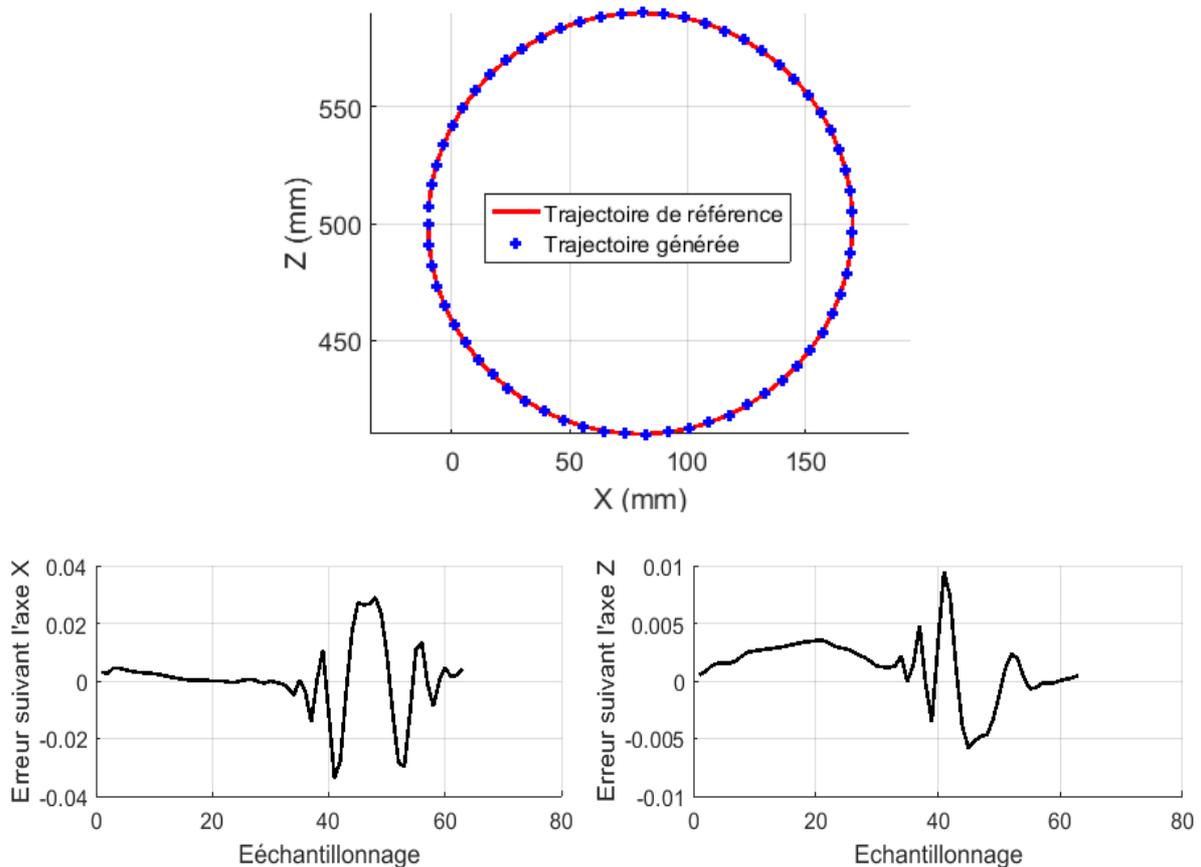


Figure 4.11 : La trajectoire désirée et générée par FOPID et les erreurs Euclidiennes entre eux.

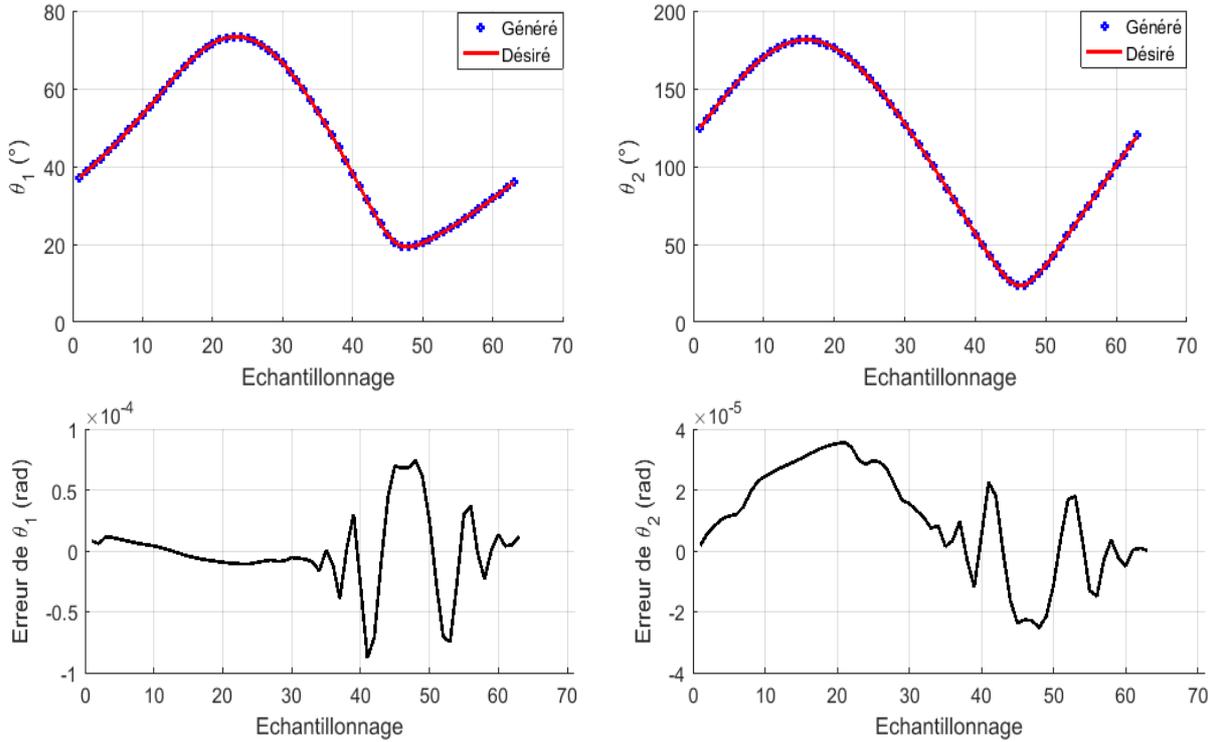


Figure 4.12 : Les angles de flexion désiré et généré et les erreurs entre eux.

4.7.2. Application 2 : implémentation de NMPC

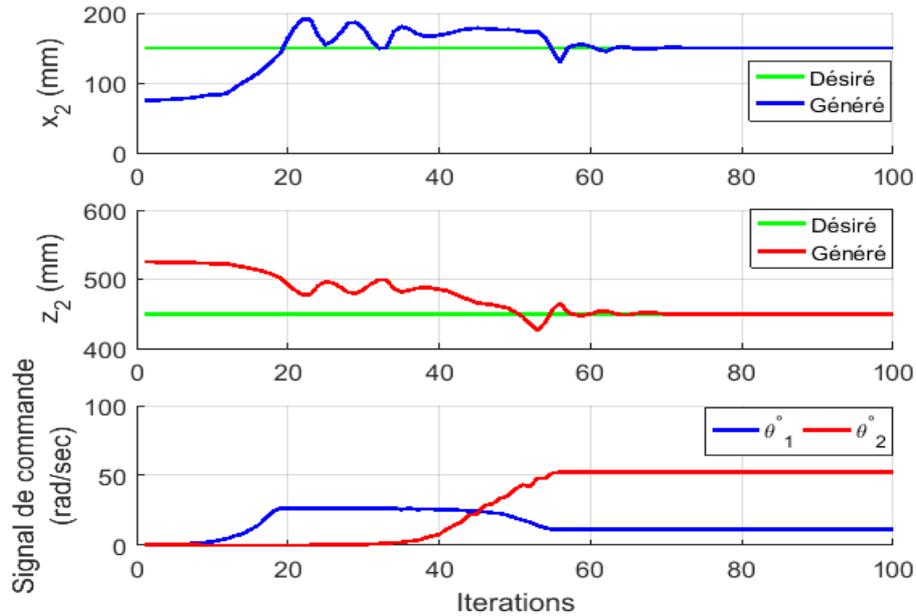
Dans cette application, l'efficacité et les performances, en termes de précision de suivi et de temps de calcul, du contrôleur NMPC basé sur les modèles cinématique et dynamique sont évaluées et comparées. Pour atteindre cet objectif, deux exemples de simulation sont effectués. Le premier exemple montre la convergence des deux contrôleurs pour atteindre une cible désirée du point final du robot (suivi d'un point de consigne). Le deuxième exemple est considéré pour implémenter les contrôleurs, basé sur les deux modèles cinématique et dynamique, pour un suivi d'une trajectoire dans un environnement Cartésien libre. Pour les deux exemples, la période d'échantillonnage est $t = 0.01$ sec et l'horizon de prédiction est $N = 2$. La fonction objective à minimiser est donnée par l'équation suivante :

$$f_{obj} = (x_{désirée} - x_{générée})^T (x_{désirée} - x_{générée}) \quad (4.46)$$

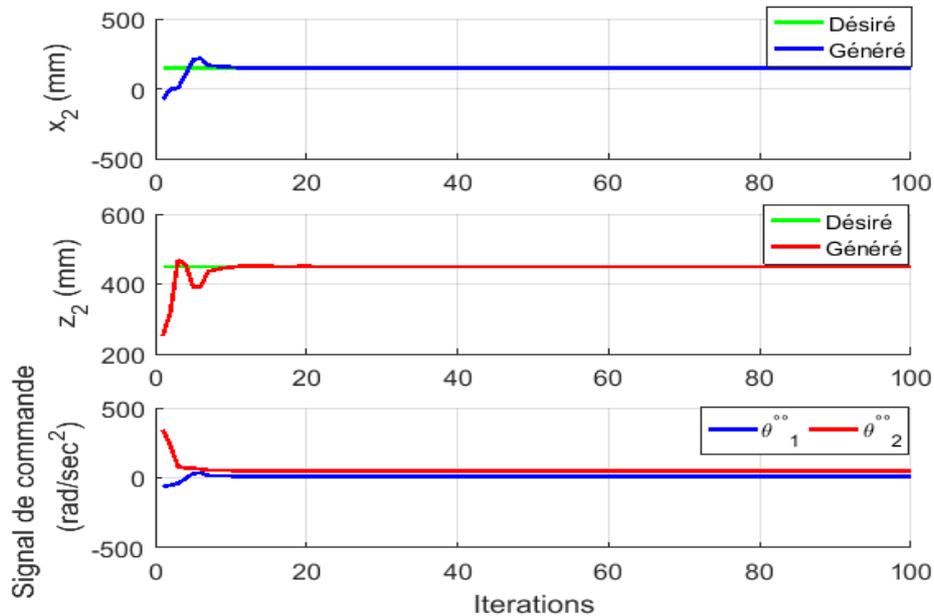
4.7.2.1. Exemple 1 : suivi d'un point de consigne

Pour les deux contrôleurs, associés aux modèles cinématique et dynamique, le robot part de la position initiale comme $[0,0,600]$ (mm) et le point de consigne est donné par $[150,0,450]$ (mm). La Figure 4.13 montre la réponse des deux contrôleurs pour atteindre l'objectif prédéfini. D'après cette Figure, on remarque que le NMPC avec le modèle

dynamique converge plus rapide que celui avec le modèle cinématique car la stabilisation du point final du robot prend moins d'itérations que le contrôleur cinématique (11 itérations contre 73 itérations). En outre le NMPC associé au modèle dynamique élimine les oscillations de l'extrémité libre du robot. Mais en termes de temps de calcul, le contrôleur associé au modèle cinématique a pris moins de temps que celui associé avec le modèle dynamique (17,34 msec contre 25,76 msec).



(a) NMPC associé au modèle cinématique.



(b) NMPC associé au modèle dynamique

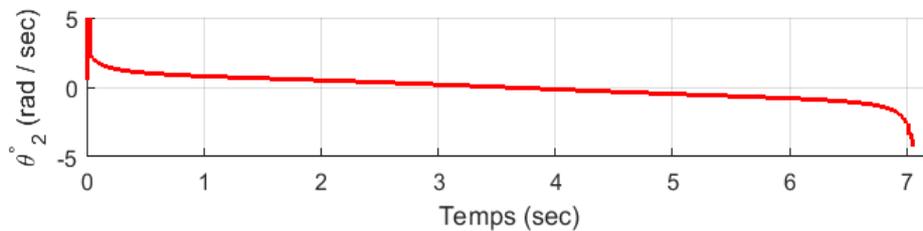
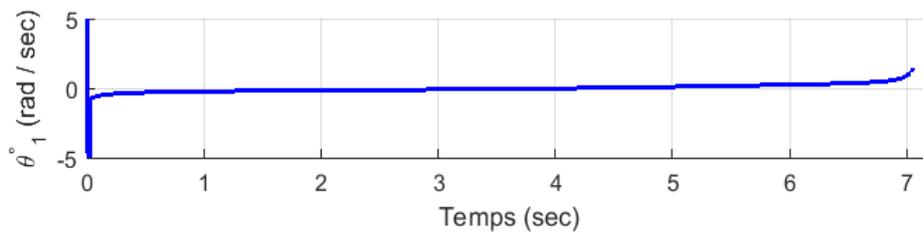
Figure 4.13 : Résultats de simulation du point de consigne évaluant la convergence du contrôleur NMPC avec les deux modèles cinématique et dynamique.

4.7.2.2. Exemple 2 : suivi d'une trajectoire

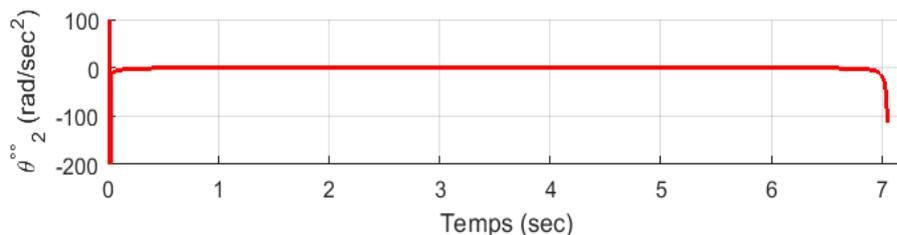
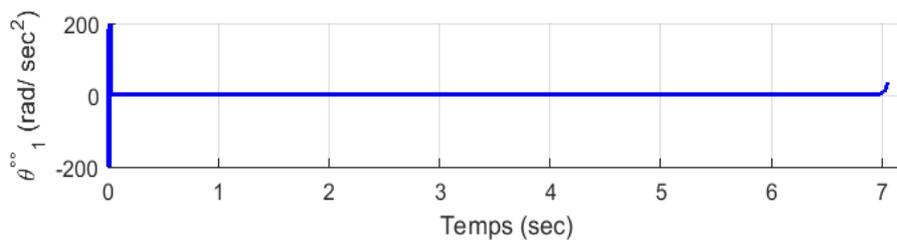
Dans cet exemple, le robot part de la position initiale comme [24.27, 0, 597.02] (mm), et la trajectoire à suivre est donnée par l'équation (4.47). La période d'échantillonnage de la commande est égale à 0,01 sec pour une durée totale du mouvement est 7 sec.

$$x_d = \begin{cases} 30\sin\left(\frac{\pi t}{5}\right) \\ 0 \\ 520 + 80\cos\left(\frac{\pi t}{5}\right) \end{cases} \quad (4.47)$$

Les résultats de simulation pour les deux contrôleurs associés avec les modèles cinématique et dynamique sont donnés sur les Figures 4.14, 4.15 et 4.16. Les signaux de commande des deux contrôleurs nécessaires pour suivre la trajectoire décrite par l'équation (4.47) sont présentés sur la Figure (4.14).



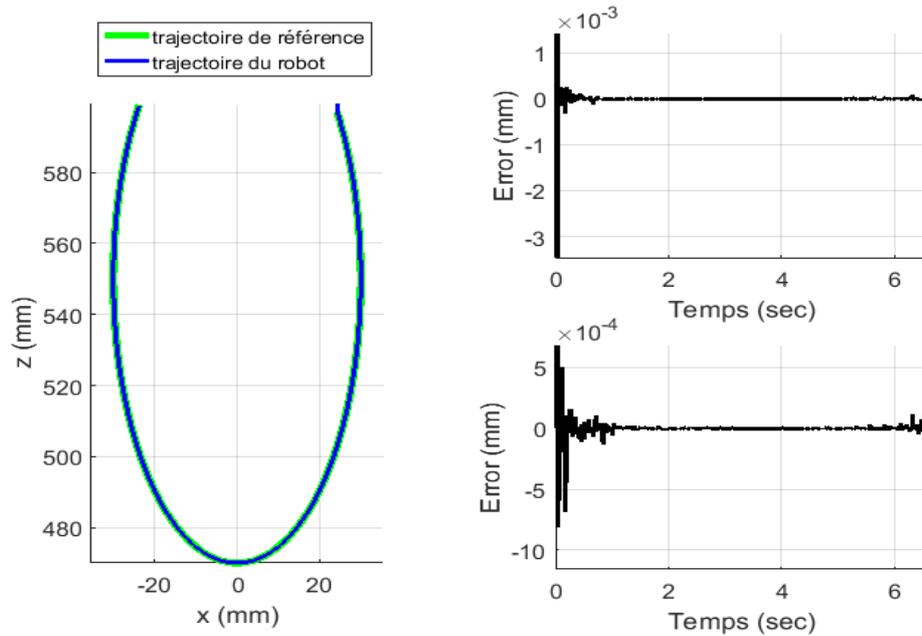
(a) NMPC associé au modèle cinématique



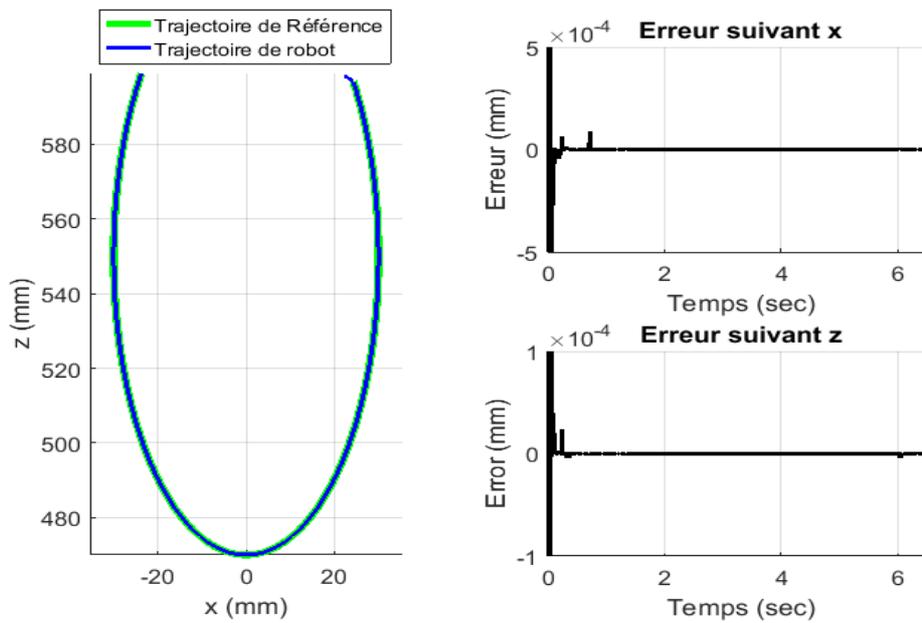
(b) NMPC associé au modèle dynamique

Figure 4.14 : Résultats de simulation du point de consigne évaluant la convergence des deux contrôleurs NMPC associés aux modèles cinématique et dynamique.

Les résultats de la simulation dans la Figure 4.15 montre le suivi de la trajectoire désirée et les erreurs Euclidiennes entre la trajectoire de référence et celle générée, par les deux contrôleurs associés au modèle cinématique et dynamique, le long des axes x et z . D'après cette Figure, on observe que les courbes sont presque superposées où les erreurs moyennes sont respectivement inférieures à 10^{-3} mm et 10^{-4} mm,



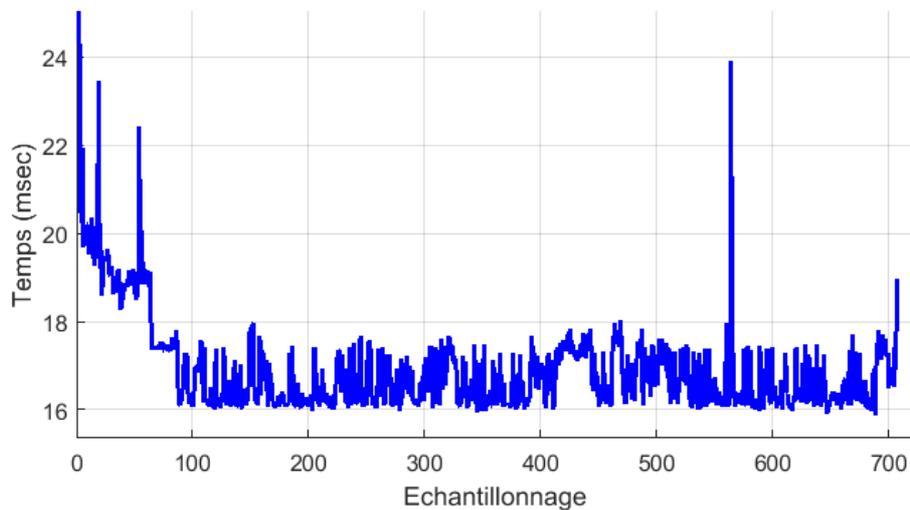
(a) NMPC avec le modèle cinématique



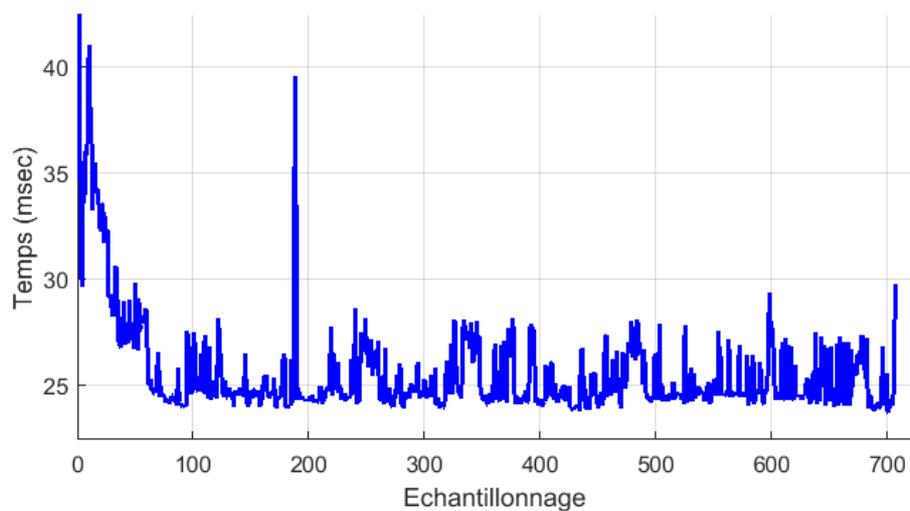
(b) NMPC avec le modèle dynamique

Figure 4.15 : Les profils des trajectoires désirée et générée et les erreurs Euclidiennes le long des axes x et z .

Le temps de calcul requis par les deux contrôleurs pour suivre la trajectoire est présenté sur la Figure 4.16. D'après cette Figure, il est clair que le temps de calcul pour le contrôleur associé au modèle cinématique est relativement faible par rapport à celle associée au modèle dynamique, où le temps de calcul moyen pour les deux contrôleurs est inférieur à 15,52 msec et 23,89 msec, respectivement. Ces résultats sont donc très encourageants pour les applications au temps réel des deux contrôleurs. Cependant, le temps de calcul dépend de la méthode d'optimisation à utiliser et du nombre de variables à optimiser ; ainsi, lorsque le nombre des sections flexibles du robot augmente, le temps de calcul augmente proportionnellement.



(a) NMPC avec le modèle cinématique



(b) NMPC avec le modèle dynamique

Figure 4.16 : Temps de la simulation du point de consigne pour évaluer la stabilisation du contrôleur NMPC avec les deux modèles cinématique et dynamique.

4.7. Comparaison des stratégies de commande

Les robots continuum ont des caractéristiques et des spécificités différentes (tailles, non-linéarités, contraintes environnementales, poids, matériaux, etc.) et ne sont généralement pas contrôlés avec les mêmes entrées (entrées électriques, câbles, entrées pneumatiques, etc.). Il est donc difficile de faire une comparaison directe entre les techniques de contrôles proposés, et cela ce fait uniquement en se basant sur les résultats obtenus. Le Tableau 4.5 liste quelques contributions sur le contrôle des robots continuum. L'aperçu des résultats de ce Tableau nous amène à dire que les contrôleurs proposés dans cette étude offrent également de bonnes performances en termes de précision et du temps de calcul.

Tableau 4.5 : Quelques contributions sur le contrôle des robots continuum.

Robot	Technique de commande	Précision	Validation
Robot continuum actionné par câbles (CDCR) : forme continue, structure rigide, contrôlée par câbles.	NMPC-PSO : Etude actuelle.	$1 \cdot 10^{-3}$ mm	Simulation
	FOPID : Etude actuelle.	$16 \cdot 10^{-3}$ mm	Simulation
	PID [13]	0.55 mm.	Simulation
	NMPC-PSO [FME 44].	$3 \cdot 10^{-3}$ mm	Simulation
	FOPID [FME 34].	$8 \cdot 10^{-3}$ mm	Simulation
	Optimized Nonlinear Sliding Mode Control [24]	0.138 mm	Simulation
Compact Bionic Handling Arm (CBHA): forme continue, structure souple, actionnement pneumatique.	Contrôleur non adaptatif par des réseaux de neurones [22].	15 mm	Simulation et expérimentale
	Contrôleur adaptatif par des réseaux de neurones [22].	5 mm	Simulation et expérimentale
	Nonadaptive support vector regressor controller [27].	11 mm	Simulation et expérimentale

	Adaptive support vector regressor controller [27].	5 mm	Simulation et expérimentale
Robots à tubes concentriques : forme continue, structure souple, actionnés par des mécanismes extérieurs à la colonne vertébrale.	NMPC [29]	$5 \cdot 10^{-4}$ mm	Simulation et expérimentale
Robot continuum actionné par tendons : forme continue, structure rigide, contrôlée par câbles.	Rétroaction statique basée sur la logique floue [24].	0.5 mm	Simulation et expérimentale
	Fuzzy-Model-Based Approach [23].	0.71 mm	Simulation et expérimentale

4.8. Conclusion

Dans ce chapitre, quatre types de contrôleurs nommément : le contrôleur PID classique, le contrôleur PID optimisé (OPID), le contrôleur PID d'ordre fractionnaire (FOPID), et la commande prédictive non linéaire (NMPC) sont évalués et comparés en termes de précision du suivi et du temps de calcul pour des points de consigne et des suivis de trajectoires. Les résultats obtenus de la comparaison des trois contrôleurs PID, OPID et FOPID a montré que le contrôleur FOPID offre des bonnes performances en termes de précision du suivi par rapport aux deux autres contrôleurs. D'autre part, les résultats de la comparaison des deux commandes NMPC associés aux modèles cinématique et dynamique a montré que le NMPC avec le modèle dynamique converge plus rapide et élimine les oscillations de l'extrémité libre du robot lors du suivi que celui associé au modèle cinématique. En revanche, le contrôleur NMPC associé au modèle cinématique a pris moins de temps de calcul que celui associé avec le modèle dynamique. En conclusion, la comparaison des résultats obtenus à partir de FOPID et NMPC proposés dans cette étude avec les travaux rapportés dans le Tableau 4.5, en termes de précision de suivi de trajectoire, il s'avère très encourageant et peuvent être utilisés pour les applications en temps réel pour ce type de robots continuum.

Conclusion générale

Le premier objectif de ce travail était de résoudre un des problèmes communs des robots continuum, qui est le modèle géométrique inverse et sa redondance. A cette fin, un algorithme est proposé. Il se compose de deux sous-algorithmes : le premier est dédié pour déterminer toutes les solutions possibles du modèle géométrique inverse pour n'importe quel point de la trajectoire à suivre par l'extrémité libre du robot, et le second est consacré à la résolution de la redondance du modèle géométrique inverse. Le second objectif est d'implémenter quelques schémas de commande afin d'assurer le suivi des trajectoires via la technique point-à-point. Afin de vérifier l'efficacité de l'algorithme développé, des exemples de simulations sont effectués. D'autre part, les quatre contrôleurs proposés ont été évalués et comparés en termes de précision de suivi et de temps de calcul. En plus, des études comparatives sont effectuées. Dans ce qui suit, les conclusions pour chaque chapitre sont données.

Dans la première partie de la thèse, une vue succincte et complète sur les robots continuum est donnée. Après avoir étudié quelques robots continuum existants ou en développement, une taxonomie des différents types de robots a été présentée afin de mettre en évidence les principales différences d'actionnement et de structure. Ensuite, une brève étude sur les méthodes et les approches les plus populaires vers la modélisation géométriques et dynamique, ainsi que les stratégies de commande les plus utilisées dans la littérature a été donnée et une évaluation qualitative de ces techniques a été présentée. Cette étude nous a permis de conclure qu'à ce jour la plupart des approches proposées dans la littérature n'ont pas abordé en profondeur le problème de redondance qui caractérise les robots continuum, et le contrôleur PID d'ordre fractionnaire n'a pas été également abordé que dans un seul travail récent.

Dans la deuxième partie, d'abord nous avons commencé par le développement du modèle géométrique direct pour une unité, une section flexible puis pour un robot continuum multi-sections. Ces modèles sont développés sur la base de l'approche couramment utilisée, nommément l'approche de l'hypothèse à courbure constante. Ensuite, nous avons présenté le schéma fonctionnel et les étapes principales de l'algorithme proposé, composé de deux sous-

algorithmes, alloué à la résolution du problème du modèle géométrique inverse et sa redondance des robots continuum. Suivi d'une étude complète et détaillée du premier sous-algorithme consacré à la détermination de toutes les solutions possibles du problème en question. Ce sous-algorithme introduit une nouvelle solution au problème, et qui permet de réduire le nombre infini de solutions à un nombre fini en prenant en compte les ensembles de poses accessibles des premières $(n - 1)$ sections flexibles, qui sont sélectionnées et créées une-par-une de la première à la $(n - 1)$ ème section flexible, dans laquelle la méthode d'optimisation par essaim des particules est adoptée pour sélectionner la première solution aléatoire du problème inverse. Ensuite, vient le problème de la redondance. Pour ce faire, le second sous-algorithme résout le problème de la redondance du modèle géométrique inverse des robots continuum, dans lequel un critère spécifique a été accordé pour sélectionner la solution appropriée parmi les variétés de solutions existantes.

Dans la troisième partie de la thèse nous avons implémenté quatre schémas de commande, nommément : Proportionnel-Intégral-Dérivé (PID), PID optimisé (OPID), PID d'Ordre Fractionnaire (FOPID) et la commande prédictive non linéaires (NMPC), afin d'assurer le suivi de trajectoire via la technique point-à-point, tant pour le modèle cinématique que dynamique du robot en question.

A la fin de chaque chapitre, des exemples de simulations sont donnés, ainsi qu'une étude comparative à d'autres approches ou techniques proposées dans la littérature est présentée. Les résultats obtenus ont démontré que l'algorithme proposé est précis et efficace sur le plan informatique pour obtenir toutes les solutions du modèle géométrique inverse et la résolution de redondance des robots continuum par rapport à certaines approches proposées dans la littérature. D'autre part, la comparaison des résultats des trois contrôleurs PID, OPID et FOPID a montré que le contrôleur FOPID offre de bonnes performances en termes de précision du suivi par rapport aux deux autres contrôleurs. Egalement, les résultats de la comparaison de deux commandes NMPC associés aux modèles cinématique et dynamique a montré que le NMPC avec le modèle dynamique converge plus rapidement et élimine les oscillations de l'extrémité libre du robot lors du suivi que celui associé au modèle cinématique. En revanche, le contrôleur NMPC associé au modèle cinématique a pris moins de temps de calcul que celui associé avec le modèle dynamique

Bien que l'algorithme proposé ne produise pas de solutions analytiques pour le modèle géométrique inverse, l'approximation numérique de la solution s'est avérée supérieure à celles obtenues par d'autres méthodes et approches, notamment la résolution de la redondance. Il est à noter que l'algorithme proposé peut être appliqué à tous les robots continuum et quel que

soit leurs nombre de sections flexibles, ce qui fait de l'algorithme un remède souverain au problème du modèle géométrique inverse de ces robots continuum.

Le travail restant à accomplir est encore important et de nouvelles perspectives et suggestions sont désormais ouvertes. Ainsi, Certaines des perspectives qui peuvent être prises en charge sont :

- Dans les modèles géométriques actuels, l'hypothèse de courbure constante est utilisée ; ainsi, un modèle plus précis reste toujours un défi et devra être développé ;
- Dans la présente étude, il a été mentionné que l'algorithme proposé est fortement lié à une méthode d'optimisation ; ainsi, la tendance à utiliser une méthode analytique est préférable. D'autre part, le temps d'exécution de l'algorithme a été ignoré, mais ce facteur est très important et devra être pris en compte dans le futur ;
- Même si certains contrôleurs répondent parfaitement aux différentes simulations, ils ne sont pas encore validés expérimentalement ; ainsi une implémentation en temps réel est nécessaire, et aussi le développement des contrôleurs adaptatifs robustes pour faire face aux différentes perturbations auxquelles sont confrontés les robots continuum.

Annexe

Les éléments de l'équation (4.47) sont donnés comme suivant [13] :

$$M_{11} = L^2 \left(\frac{4I_x}{L^2} - \frac{\theta_1\theta_2}{3} + \frac{\theta_1\theta_2^3}{36} + \frac{\theta_1^3\theta_2}{60} - \frac{2\theta_1^2}{9} + \frac{\theta_1^4}{144} - \frac{\theta_2^2}{2} + \frac{\theta_2^4}{45} + \frac{\theta_1^2\theta_2^2}{36} - \frac{\theta_1^2\theta_2^4}{720} - \frac{\theta_1^3\theta_2^3}{720} - \frac{\theta_1^4\theta_2^2}{1080} + \frac{\theta_1^4\theta_2^4}{2160} + 5 \right), \quad (\text{A. 1})$$

$$M_{12} = L^2 \left(\frac{2I_x}{L^2} - \frac{\theta_1\theta_2}{9} + \frac{\theta_1\theta_2^3}{90} + \frac{\theta_1^3\theta_2}{180} - \frac{\theta_1^2}{24} + \frac{\theta_1^4}{720} - \frac{5\theta_2^2}{24} + \frac{7\theta_2^4}{720} + \frac{\theta_1^2\theta_2^2}{96} - \frac{\theta_1^2\theta_2^4}{1728} - \frac{\theta_1^3\theta_2^3}{1800} - \frac{\theta_1^4\theta_2^2}{2880} + \frac{\theta_1^4\theta_2^4}{51840} + \frac{3}{2} \right), \quad (\text{A. 2})$$

$$M_{21} = L^2 \left(\frac{2I_x}{L^2} - \frac{\theta_1\theta_2}{9} + \frac{\theta_1\theta_2^3}{90} + \frac{\theta_1^3\theta_2}{180} - \frac{\theta_1^2}{24} + \frac{\theta_1^4}{720} - \frac{5\theta_2^2}{24} + \frac{7\theta_2^4}{720} + \frac{\theta_1^2\theta_2^2}{96} - \frac{\theta_1^2\theta_2^4}{1728} - \frac{\theta_1^3\theta_2^3}{1800} - \frac{\theta_1^4\theta_2^2}{2880} + \frac{\theta_1^4\theta_2^4}{51840} + \frac{3}{2} \right), \quad (\text{A. 3})$$

$$M_{22} = L^2 \left(\frac{\theta_2^4}{1440} - \frac{\theta_2^2}{36} + \frac{2I_x}{L^2} + \frac{1}{2} \right), \quad (\text{A. 4})$$

$$C_{11} = L^2 \left(\frac{\theta_1^3\theta_2^4}{16200} - \frac{\theta_1^3\theta_2^2}{1080} - \frac{\theta_1^3}{360} - \frac{\theta_1^2\theta_2^3}{800} + \frac{\theta_1^2\theta_2}{120} - \frac{\theta_1\theta_2^4}{1080} + \frac{\theta_1\theta_2^2}{72} + \frac{\theta_1}{18} + \frac{\theta_2^3}{120} - \frac{\theta_2}{18} \right), \quad (\text{A. 5})$$

$$C_{12} = L^2 \left(\frac{\theta_1^4\theta_2^3}{6480} - \frac{\theta_1^4\theta_2}{720} - \frac{\theta_1^3\theta_2^2}{300} + \frac{\theta_1^3}{90} - \frac{\theta_1^2\theta_2^3}{216} + \frac{\theta_1^2\theta_2}{24} + \frac{\theta_1\theta_2^2}{15} - \frac{2\theta_1}{9} + \frac{7\theta_2^3}{90} - \frac{5\theta_2}{6} \right), \quad (\text{A. 6})$$

$$C_{13} = L^2 \left(\frac{\theta_1^4\theta_2^3}{12960} - \frac{\theta_1^4\theta_2}{1440} - \frac{\theta_1^3\theta_2^2}{600} + \frac{\theta_1^3}{180} - \frac{\theta_1^2\theta_2^3}{432} + \frac{\theta_1^2\theta_2}{48} + \frac{\theta_1\theta_2^2}{30} - \frac{\theta_1}{9} + \frac{7\theta_2^3}{180} - \frac{5\theta_2}{12} \right), \quad (\text{A. 7})$$

$$C_{21} = L^2 \left(-\frac{\theta_1^4 \theta_2^3}{10800} + \frac{\theta_1^4 \theta_2}{1080} + \frac{\theta_1^3 \theta_2^4}{12960} + \frac{\theta_1^3 \theta_2^2}{1440} - \frac{\theta_1^3}{360} + \frac{\theta_1^2 \theta_2^3}{900} - \frac{\theta_1^2 \theta_2}{90} - \frac{\theta_1 \theta_2^4}{864} - \frac{\theta_1 \theta_2^2}{48} + \frac{\theta_1}{12} - \frac{\theta_2^3}{30} + \frac{7\theta_2}{18} \right), \quad (\text{A. 8})$$

$$C_{22} = 0, \quad (\text{A. 9})$$

$$C_{23} = L^2 \left(\frac{\theta_2^3}{720} - \frac{\theta_2}{36} \right), \quad (\text{A. 10})$$

$$K_1 = g.m.L \left(\frac{5\theta_1}{3} + \frac{\theta_2}{2} - \frac{\theta_1 \theta_2^2}{6} - \frac{\theta_1^2 \theta_2}{4} + \frac{\theta_1 \theta_2^4}{120} - \frac{7\theta_1^3}{30} - \frac{\theta_2^3}{24} + \frac{\theta_1^2 \theta_2^3}{48} - \frac{\theta_1^3 \theta_2^2}{36} - \frac{\theta_1^3 \theta_2^4}{720} + \frac{\theta_1^3 \theta_2^2}{18} \right) + \frac{EI_b \theta_1}{L}, \quad (\text{A. 11})$$

$$K_2 = g.m.L \left(\frac{\theta_1}{2} + \frac{\theta_2}{3} - \frac{\theta_1 \theta_2^2}{8} - \frac{\theta_1^2 \theta_2}{6} + \frac{\theta_1^4 \theta_2}{72} - \frac{\theta_1^3}{12} - \frac{\theta_2^3}{30} + \frac{\theta_1^2 \theta_2^3}{60} + \frac{\theta_1^3 \theta_2^2}{48} - \frac{\theta_1^4 \theta_2^3}{720} \right) + \frac{EI_b \theta_2}{L}, \quad (\text{A. 12})$$

Références bibliographiques

- [1] B. Singh, N. Sellappan, and P. Kumaradhas, “Evolution of industrial robots and their applications,” *International Journal of Emerging Technology and Advanced Engineering*, Vol. 3, No. 5, pp. 763–768, (2013).
- [2] J. Burgner-Kahrs, D. C. Rucker and H. Choset, “Continuum robots for medical applications: a survey,” *IEEE Trans Rob*, Vol. 31, No. 6, pp. 1261–1280, (2015).
- [3] Y. Zhang and M. Lu, “A review of recent advancements in soft and flexible robots for medical applications,” *The International Journal of Medical Robotics and Computer Assisted Surgery*, Vol. 16, no. 3, p.e2096, (2020).
- [4] Chen, F. and M. Y. Wang, “Design optimization of soft robots: A review of the state of the art,” *IEEE Robotics & Automation Magazine*, Vol. 27, No. 4, pp.27–43, (2020).
- [5] L. S. Cowan and I. D. Walker, “The importance of continuous and discrete elements in continuum robots,” *International Journal of Advanced Robotic Systems*, Vol. 10, No. 3, p. 165, (2013).
- [6] G. Bao, S. Cai, Z. Wang et al., “Flexible pneumatic robotic actuator FPA and its applications,” in *Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 867–872, Shenzhen, China, December 2013.
- [7] M. Russo, S. Mohammad H. Sadati, et al., “Continuum Robots: An overview,” *Advanced Intelligent systems*, Vol. 5, No. 5, Article ID: 2200367, (2023).
- [8] H. D. Yang and A. T. Asbeck, “A new manufacturing process for soft robots and soft/rigid hybrid robots,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8039–8046, Madrid, Spain, October 2018.
- [9] D. A. Troncoso, J. A. Robles-Linares, M. Russo, et al., “Continuum robot for remote applications: from industrial to medical surgery with slender continuum robots,” *IEEE Robot. Autom*, pp. 2–13, (2022).

- [10] T. Morales Bieze, “Contribution to the Kinematic Modeling and Control of Soft Manipulators using Computational Mechanics,” Ph. D. dissertation, University of Sciences and Technologies of Lille, (2017)
- [11] X. Dong, M. Wang, A. Mohammad, et al., “Continuum robots collaborate for safe manipulation of high-temperature flame to enable repairs in challenging environments,” *IEEE/ASME Trans. Mechatron*, Vol. 27, pp. 4217–4220, (2022).
- [12] J. Zhang, Q. Fang, P. Xiang, et al., “A survey on design, actuation, modeling, and control of continuum robot,” *Cyborg and Bionic Systems*, Vol 2022, Article ID: 9754697, (2022).
- [13] A. Amouri, “Contribution la modélisation dynamique d'un robot flexible bionique,” Ph. D. dissertation, Frères Mentouri University, Constantine 1, Algeria, July (2017).
- [14] G. Robinson and J. B. C. Davies, “Continuum robots-a state of the art,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, Detroit, MI, USA, Vol. 4, pp. 2849–2854, (1999).
- [15] W. McMahan, V. Chitrakaran, M. Csencsits, et al., “Field trials and testing of the OctArm continuum manipulator,” in *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 2336 –2341, (2006).
- [16] W. McMahan, B. A. Jones, and I. D. Walker, “Design and implementation of a multi-section continuum robot: Air-Octor,” in *Proceedings of International Conference on Intelligent Robots and Systems*, pp. 2578 –2585, (2005).
- [17] C. Laschi, B. Mazzolai, V. Mattoli, et al., “Design of a biomimetic robotic Octopus arm,” *Bioinspiration & Biomimetics*, Vol. 4 (1): 015006, (2009).
- [18] V. Anderson, and R. Horn, “Tensor arm manipulator,” U.S. Patent 3-497-083, (1970).
- [19] P. Sabetian, A. Feizollahi, F. Cheraghpour, and S. A. A. Moosavian, “A compound robotic hand with two under-actuated fingers and a continuous finger,” in *Proceedings of the IEEE International Symposium on Safety, Security, and Rescue Robotics*, Kyoto, Japan, pp. 238–244, June (2011).
- [20] R. Mutlu, S. K. Yildiz, G. M. Alici, and G. M. Spinks, “Mechanical stiffness augmentation of a 3D printed soft prosthetic finger,” in *Proceedings of the IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, Banff, Alberta, Canada, pp. 7–12, July (2016).
- [21] I. P. Georgilas and V. D. Tourassis, “Ermis-a novel biologically inspired flexible robotic mechanism for industrial applications,” in *Proceedings of the IEEE/ASME*

- International Conference on Advanced Intelligent Mechatronics, pp. 1504–1509, Singapore, July (2009).
- [22] W. S. Rone and P. Ben-Tsvi, “Continuum robotic tail loading analysis for mobile robot stabilization and maneuvering,” in Proceedings of the ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, Buffalo, NY, USA, pp. 1–8, August (2014).
- [23] Z. Zheng, K. Wang, X. Wu, Q. Zhang, and K. Li, “Dynamic analysis of elastic projecting robot inspired by chameleon tongue,” in Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO), Zurich, Switzerland, pp. 2088–2094, December (2018).
- [24] T. Mahl, A. Hildebrandt, and O. Sawodny, “A variable curvature continuum kinematics for kinematic control of the bionic handling assistant,” IEEE Transactions on Robotics, Vol. 30, No. 4, (2014).
- [25] R. Buckingham, “Snake-arm robot,” Industrial Robot: An International Journal, Vol. 29, No. 3, pp. 242–245, (2002).
- [26] H. Wang, S. Fan, F. Ni, and H. Liu, “Biologically inspired guidelines for the design of the hyper-dexterous manipulator,” in Proceedings of the 2016 IEEE International Conference on Mechatronics and Automation, Harbin, China, pp. 641–646, August (2016).
- [27] S. Kolachalama, S. Lakshmanan, ”Continuum robots for manipulation applications: A survey,” Journal of Robotics, Vol. 2020, Article ID 4187048, pp. 1–19, (2020).
- [28] A. Amouri, A. Cherfia, A. Belkhiri, et al. “Bio-inspired a novel dual-cross-module sections cable-driven continuum robot: design, kinematics modeling and workspace analysis,” Journal of the Brazilian Society of Mechanical Sciences and Engineering, Vol. 45, No. 265, (2023).
- [29] K. T. Song and H. C. Tsai, “Visual servoing and compliant motion control of a continuum robot,” in Proceedings of the 18th International Conference on Control, Automation and Systems (ICCAS), Pyeong Chang, South Korea, pp. 734–739, October (2018).
- [30] M. M. Coad, L. H. Blumenschein, S. Cutler, et al. “Vine robots: design, teleoperation, and deployment for navigation and exploration,” Proceeding IEEE Robotics & Automation Magazine, Vol. 27, No.3, pp. 120–132. (2020).

- [31] M. B. Wooten and I. D. Walker, “A novel vine-like robot for in-orbit inspection,” in Proceedings of the 45th International Conference on Environmental Systems, Bellevue, WA, USA, pp. 1–11, July (2015).
- [32] M. Wooten, C. Frazelle, I. D. Walker, et al. “Exploration and inspection with vine-inspired continuum robots,” in Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, pp. 1–5, May (2018).
- [33] M. Wooten and I. D. Walker, “Vine-inspired continuum tendril robots and circumnutations,” *Robotics*, Vol. 7, No. 3, p. 58, (2018).
- [34] I. D. Walker, “Continuous Backbone "Continuum" Robot Manipulators,” *ISRN Robotics*, pp.1–19, (2013).
- [35] A. Bajo, N. Simaan, “Hybrid motion/force control of multi-backbone continuum robots,” *The International Journal of Robotics Research*, Vol. 35, No. 4, pp. 422–434, (2016).
- [36] D. C. Rucker, Bryan A. Jones and R. J. Webster, III, “A Geometrically exact model for externally loaded concentric-tube continuum robots,” *IEEE Trans Robot.* Vol. 26, No. 5, pp. 769–780, (2011).
- [37] Q. Fei, J. Feng, D. Bai, W. Yaoyao, et al., “Kinematic analysis and navigation method of a cable-driven continuum robot used for minimally invasive surgery,” *The International Journal of Medical Robotics and Computer Assisted Surgery*, Vol. 15(4) : e2007, (2007).
- [38] M. M. Dalvand, S. Nahavandi, R. D. Howe, “General forward kinematics for tendon-driven continuum robots,” *IEEE Access*, Vol. 10, pp. 60330–60340. (2022).
- [39] S. Mbakop, G. Tagne, M. H. Frouin, et al., “Inverse dynamics model-based shape control of soft continuum finger robot using parametric curve,” *IEEE Robotics and Automation*, Vol. 6, No. 4, pp. 8053–8060, (2021).
- [40] D. Bruder, A. Sedal, R. Vasudevan, and C. D. Remy, “Force generation by parallel combinations of fiber-reinforced fluid-driven actuators,” *IEEE Robotics and Automation Letters*, Vol. 3, No. 4, pp. 3999–4006, (2018).
- [41] T. A. Q. I. Abrar, F. Putzu, J. Konstantinova, and K. Althoefer, “EPAM: eversive pneumatic artificial muscle,” in Proceedings of the IEEE International Conference on Soft Robotics (RoboSoft), pp. 19–24, Seoul, South Korea, April 2019.
- [42] G.S. Chirikjian, “Conformational modeling of continuum structures in robotics and structural biology: A review,” *Advanced Robotics*, Vol. 29, No. 13, pp. 817–829, (2015).

- [43] B. A. Jones, W. McMahan, and I. D. Walker, “Design and analysis of a novel pneumatic manipulator,” *IFAC Proceedings*, Vol. 37, No. 14, pp. 687–692, (2004).
- [44] S. Xu, G. Li, D. Song, L. Sun, and J. Liu, “Real-time shape recognition of a deformable link by using a self-organizing map,” in *Proceedings of the IEEE 14th International Conference on Automation Science and Engineering (CASE)*, pp. 586–591, Munich, Germany, August (2018).
- [45] H. Mao, J. Santoso, C. Onal, and J. Xiao, “Sim-to-real transferable object classification through touch-based continuum manipulation,” in *Proceedings of the International Symposium on Experimental Robotics (ISER)*, Springer, Buenos Aires, Argentina, November (2018).
- [46] B. A. Jones and I. D. Walker, “Practical kinematics for realtime implementation of continuum robots,” *IEEE Transactions on Robotics*, Vol. 22, No. 6, pp. 1087–1099, (2006).
- [47] Z. Zheng, K. Wang, X. Wu, Q. Zhang, and K. Li, “Dynamic analysis of elastic projecting robot inspired by chameleon tongue,” in *Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 2088–2094, Zurich, Switzerland, December (2018).
- [48] G. Niu, J. Wang, and K. Xu, “Model analysis for a continuum aircraft fuel tank inspection robot based on the Rzeppa universal joint,” *Advances in Mechanical Engineering*, Vol. 10, No. 5, (2018).
- [49] I. P. Georgilas and V. D. Tourassis, “Ermis-a novel biologically inspired flexible robotic mechanism for industrial applications,” in *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pp. 1504–1509, Singapore, July (2009).
- [50] N. Guochen, W. Li, G. Qingji, and H. Dandan, “Pathtracking algorithm for aircraft fuel tank inspection robots,” *International Journal of Advanced Robotic Systems*, Vol. 11, No. 5, p. 82, (2014).
- [51] A. Parvaresh, S. A. Moosavi, and S. A. A. Moosavian, “Identification and position control of a continuum robotic arm,” in *Proceedings of the 2017 5th RSI International Conference on Robotics and Mechatronics (ICRoM)*, pp. 310–315, Tehran, Iran, October (2017).
- [52] W. S. Rone and P. Ben-Tsvi, “Continuum robotic tail loading analysis for mobile robot stabilization and maneuvering,” in *Proceedings of the ASME International*

- Design Engineering Technical Conferences & Computers and Information in Engineering Conference, pp. 1–8, Buffalo, NY, USA, August (2014).
- [53] T. Yukisawa, S. Nishikawa, R. Niiyama, et al., “Ceiling continuum arm with extensible pneumatic actuators for desktop workspace,” in Proceedings of the 2018 IEEE International Conference on Soft Robotics (RoboSoft), pp. 196–201, Livorno, Italy, April (2018).
- [54] Z. Li and R. Du, “Design and implementation of a biomimetic wire-driven underactuated serpentine manipulator,” *Transaction on Control and Mechanical Systems*, Vol. 1, pp. 250–258, (2012).
- [55] M. Eder, M. Karl, A. Knoll, et al., “Continuum worm-like robotic mechanism with decentral control architecture,” in Proceedings of the 2014 IEEE International Conference on Automation Science and Engineering (CASE), pp. 866–871, Taipei, Taiwan, August (2014).
- [56] K. Xu, Z. Liu, B. Zhao, et al., “Composed continuum mechanism for compliant mechanical postural synergy: an anthropomorphic hand design example,” *Mechanism and Machine theory*, Vol. 132, pp. 108–122, (2019).
- [57] F. Putzu, T. Abrar, and K. Althoefer, “Plant-inspired soft pneumatic eversion robot,” in Proceedings of the IEEE International Conference on Biomedical Robotics and Biomechatronics (Biorob), Enschede, the Netherlands, pp. 1327–1332, August (2018).
- [58] J. S. Mehling, M. A. Diftler, M. Chu, et al., “A minimally invasive tendril robot for in-space inspection,” in Proceedings of the IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics. BioRo, pp. 690–695, Pisa, Italy, February (2006).
- [59] A. D. Marchese, R. K. Katzschmann, and D. Rus, “Whole arm planning for a soft and highly compliant 2d robotic manipulator,” in Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, pp. 554–560, September (2014).
- [60] R. J. Webster, B. A. Jones, “Design and kinematic modeling of constant curvature continuum robots: a review,” *International Journal of Constant Curvature Continuum Robots*, Vol. 29, No. 13, pp. 1661–1683, (2010).
- [61] A. Garriga-Casnovas, Y. Rodriguez, F. Baena, “Kinematics of continuum robots with constant curvature bending and extension capabilities,” *Journal of Mechanisms and Robotics*, Vol. 11(1): 011010 (12 pages), (2019).

- [62] S. Mosqueda, Y. Moncada, C. Murrugarra, et al., “Constant curvature kinematic model analysis and experimental validation for tendon driven continuum manipulators,” In Proceedings of the 15th International Conference on Informatics in Control, Automation and Robotics, Vol. 2, pp. 211–218, (2018).
- [63] A. Amouri, “Investigation of the constant curvature kinematic assumption of a 2-Dofs cable-driven continuum robot,” UPB Scientific Bulletin, Series D: Mechanical Engineering, Vol. 81, pp. 27–38, (2019).
- [64] P. Zhou, J. Yao, S. Zhang et al. (2022) “A bioinspired fishbone continuum robot with rigid-flexible-soft coupling structure,” *Bioinspiration Biomimetics*, Vol. 17, pp. 066012, (2022).
- [65] A. Merrad, A. Amouri, A. Cherfia, et al. “A reliable algorithm for obtaining all-inclusive inverse kinematics’ solutions and redundancy resolution of continuum robots,” *Arabian Journal for Science and Engineering*, Vol. 48, pp. 3351–3366, (2023).
- [66] M. W. Hannan and I.D. Walker, “The elephant trunk manipulator, design and implementation,” *IEEE/ASME International Conference on Advanced Intelligent Mechatronics Proceedings*, Como, Italy, pp. 14–19, July (2001).
- [67] M. W. Hannan and I.D. Walker, “Kinematics and the implementation of an elephant’s trunk manipulator and other continuum style robots,” *Journal of Robotic Systems*, Vol. 20, No. 2, pp. 45–63, (2003).
- [68] M. K. Mishra, A. K. Samantaray, G. Chakraborty, et al., “Kinematics model of bionic manipulator by using elliptic integral approach,” *Machines, Mechanism and Robotics*, Lecture Notes in Mechanical Engineering, Springer, Singapore (2022).
- [69] R. J. Webster III, J. M. Romano, and N. J. Cowan, “Mechanics of precurved-tube continuum robots,” *IEEE Transactions on Robotics*, Vol. 25, No. 1, pp. 67–78, (2009).
- [70] I. S. Godage, G. A. Medrano Cerda, D. T. Branson, et al., “Modal kinematics for multisession continuum arms,” *Bioinspiration and Biomimetics*, Vol. 10, pp. 1–20, (2015).
- [71] I.S. Godage, E. Guglielmino, D. T. Branson, “Novel modal approach for kinematics of multisection continuum arms,” *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, San Francisco, CA, USA, pp.1093–1098, (2011).
- [72] S. Huang, D. Meng, X. Wang, B. Liang, et al, “A 3D static modeling method and experimental verification of continuum robots based on pseudo-rigid body theory,”

- In: IEEE International Conference on Intelligent Robots and Systems, Macau, China, (2019).
- [73] I. Singh, Y. Amara, A. Melingui, et al., “Modeling of continuum manipulators using pythagorean hodograph curves,” *Soft Robot*, Vol. 5, No. 4, pp. 425–442, (2018).
- [74] D. Trivedi, A. Lotfi, C. Rahn, “Geometrically exact models for soft robotic manipulators,” *IEEE Transactions on Robotics*, Vol. 24, No. 4, pp. 773–780, (2008).
- [75] S. Neppalli, M.A. Csencsits, B. A. Jones, et al., “Closed form inverse kinematics for continuum manipulators,” *Advanced Robotics*, Vol. 23, pp. 2077–2091, (2009).
- [76] G. S. Chirikjian, J. W. Burdick, “A modal approach to hyper-redundant manipulator kinematics,” *IEEE Transactions on Robotics*, Vol. 10, No. 3, pp. 343–354, (1994).
- [77] B. Zhao, L. Zeng, B. Wu, et al., “A continuum manipulator with closed-form inverse kinematics and independently tunable stiffness,” In: *IEEE International Conference on Robotics and Automation (ICRA)*, pp.1847–1853, (2020).
- [78] A. Garriga-Casanovas, Y. Rodriguez, F. Baena, “Kinematics of continuum robots with constant curvature bending and extension capabilities,” *Journal of Mechanisms and Robotics*, (2019).
- [79] T. C. Hu, A. B. Kahng, G. Robins, “Optimal robust path planning in general environments,” *IEEE Transactions on Robotics and Automation*, Vol. 9, No. 6, pp. 775–784, (1993).
- [80] F. Fahimi, H. Ashrafiuon, C. Nataraj, “An improved inverse kinematic and velocity solution for spatial hyper-redundant robots,” *IEEE Transactions on Robotics and Automation*, Vol. 18, No.1, pp.103–107, (2002).
- [81] A. Melingui, O. Lakhal, B. Daachi, et al., “Adaptive neural network control of a compact bionic handling arm,” *IEEE/ASME Transactions on Mechatronics*, Vol. 20, No. 6, pp. 2862–2875, (2015).
- [82] O. Lakhal, A. Melingui, R. Merzouki, “Hybrid approach for modeling and solving of kinematics of a compact bionic handling assistant manipulator,” *IEEE/ASME Transactions on Mechatronics*, Vol. 21, No. 3, pp. 326–1335, (2016).
- [83] A. Amouri, C. Mahfoudi, A. Zaatari, et al., “A metaheuristic approach to solve inverse kinematics of continuum manipulators,” *Journal of Systems and Control Engineering*, Vol. 231, No. 5, pp. 380–394, (2017).
- [84] O. Lakhal, A. Melingui, A. Chibani, et al., “Inverse kinematic modeling of a class of continuum bionic handling arm,” In: *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Besacon*, pp. 337–1342, (2014).

- [85] T. Mahl, T. A. Hildebrandt, O. Sawodny, “A variable curvature continuum kinematics for kinematic control of the bionic handling assistant,” *IEEE Transactions on Robotics*, Vol. 30, No. 4, pp. 935–949, (2014).
- [86] I. Singh, O. Lakhal, Y. Amara, et al., “Performances evaluation of inverse kinematic models of a compact bionic handling assistant,” In: *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Macau, pp. 264–269, (2017).
- [87] B. He, Z. Wang, Q. Li, et al., “An analytic method for the kinematics and dynamics of a multiple-backbone continuum robot,” *International Journal of Advanced Robotic Systems*, Vol. 10, pp. 1–13, (2013).
- [88] V. Falkenhahn, T. Mahl, A. Hildebrandt, et al., “Dynamic modeling of bellows-actuated continuum robots using the Euler–Lagrange formalism,” *IEEE Transactions on Robotics*, Vol. 31, No. 6, pp. 1483–1496, (2015).
- [89] A. Amouri, A. Zaatri, and C. Mahfoudi, “Dynamic modeling of a class of continuum manipulators in fixed orientation,” *Journal of Intelligent & Robotic Systems*, Vol. 91, No. 3, pp. 413–424, (2018).
- [90] L. Bousbia, A. Amouri, A. Cherfia, “Dynamics modeling of a 2-DoFs cable-driven continuum robot,” *World Journal of Engineering*, (2022).
- [91] A. Amouri, C. Mahfoudi, and A. Zaatri, “Dynamic Modeling of a Spatial Cable-Driven Continuum Robot Using Euler-Lagrange Method,” *International Journal of Engineering and technology innovation*, Vol. 10, No. 1, pp. 60–74, Jan (2020).
- [92] W. S. Rone and P. Ben-Tzvi, “Continuum robot dynamics Utilizing the principle of virtual power,” *IEEE Transactions on Robotics*, Vol. 30, No. 1, pp. 275–287, (2014).
- [93] W. S. Rone and P. Ben-Tzvi, “Mechanics modeling of multi-segment rod-driven continuum robots,” *Journal of Mechanisms and Robotics*, Vol. 6, No. 4, (2014).
- [94] I. A. Gravagne, C. D. Rahn, I. D. Walker, “Large deflection dynamics and control for planar continuum robots,” *IEEE/ASME Transactions on Mechatronics*, Vol. 8, No. 2, pp. 299–307, (2003).
- [95] D. C. Rucker and R. J. Webster III, “Statics and dynamics of continuum robots with general tendon routing and external loading,” *IEEE Transactions on Robotics*, Vol. 27, No. 6, pp. 1033–1044, (2011).
- [96] J. Till, V. Aloï, C. Rucker, “Real-time dynamics of soft and continuum robots based on cosserat rod models,” *The International Journal of Robotics Research*, Vol. 38, No. 6, pp. 723–746, (2019).

- [97] A. Amouri, A. Cherfia, H. Merabti, et al., “Nonlinear model predictive control of a class of continuum robots using kinematic and dynamic models,” *FME Transactions*, Vol. 50, No. 2, pp. 339–350, (2022).
- [98] A. Amouri, H. Merabti, A. Cherfia, et al., “Nonlinear model predictive control for trajectory tracking of a class of continuum robots,” *UPB Scientific Bulletin, Series D: Mechanical Engineering*, Vol. 84, No. 3, pp. 19–32, (2022).
- [99] A. Ghouli, K. Kara, M. Benrabah, et al., “Optimized nonlinear sliding mode control of a continuum robot manipulator,” *Journal of Control, Automation and Electrical Systems*, Vol. 33, pp. 1355–1363, (2022).
- [100] A. Belkhiri, A. Amouri and A. Cherfia, “Design of fractional-order PID controller for trajectory tracking control of continuum robots,” *FME Transactions Journal*, Vol. 51, No. 2, pp. 243–252, (2023).
- [101] P. Qi, C. Liu, A. Ataka, et al., “Kinematic control of continuum manipulators using a fuzzy-model-based approach,” *IEEE Transactions on Industrial Electronics*, Vol. 63, No. 8, pp. 5022–5035, (2016).
- [102] W. Ba, X. Dong, A. Mohammad, “Design and validation of a novel fuzzy-logic-based static feedback controller for tendon-driven continuum robots,” *IEEE/ASME Transactions on Mechatronics*, pp. 3010–3021, (2021).
- [103] P. Qi, C. Liu, L. Zhang, “Fuzzy logic control of a continuum manipulator for surgical applications,” in: *Proceedings of IEEE International Conference on Robotics, Biomimetics*, pp. 413–418, (2014).
- [104] M. Goharimanesh, A. Mehrkish and F. Janabi-Sharifi, “A fuzzy reinforcement learning approach for continuum robot Control,” *Journal of Intelligent and Robotic Systems*, Vol. 100, pp. 809–826, (2020).
- [105] A. Melingui, J. J. Mvogo Ahanda, O. Lakhal, “Adaptive algorithms for performance improvement of a class of continuum manipulators,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, Vol. 48, No. 9, pp. 1531–1541, (2018).
- [106] D. E. Goldberg, “Genetic algorithms in search, optimization and machine learning,” Reading, MA: Addison-Wesley, (1989).
- [107] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in: *Proceedings of IEEE, International Conference on Neural Networks (ICNN)*, Vol. 4, pp. 1942–1948, (1995).

- [108] Y. Shi, R. Eberhart, “A modified particle swarm optimizer,” in: IEEE International Conference on Evolutionary Computation Proceedings, IEEE World Congress on Computational Intelligence, Anchorage, AK, USA, pp. 69–73, (1998).
- [109] R. C. Eberhart, Y. Shi, “Comparing inertia weights and constriction factors in particle swarm optimization,” in: Congress of Evolutionary Computation, San Diego, CA. 1, pp. 84–88, (2000).
- [110] M. Rolf, J. J. Steil, “Efficient exploratory learning of inverse kinematics on a bionic elephant trunk,” IEEE Transactions on Neural Networks and Learning Systems, Vol. 25, No. 6, pp. 1147–1160, (2014).