

ÉCOLE DOCTORALE MSII

Laboratoire ICube (UMR 7357)

THÈSE présentée par :
Guillaume LODS

Soutenue le :
10 Décembre 2024

Pour obtenir le grade de :
Docteur de l'Université de Strasbourg

Discipline/Spécialité :
Robotique

**Modélisation et utilisation de la
compliance des robots continus
Application aux robots à tubes concentriques**

THÈSE DIRIGÉE PAR :

Mr BAYLE Bernard

Professeur, Université de Strasbourg

Mr NAGEOTTE Florent

Professeur, Université de Strasbourg

THÈSE ENCADRÉE PAR :

Mr ROSA Benoit

Chargé de recherche, CNRS, Strasbourg

RAPPORTEURS :

Mr LAURENT Guillaume

Professeur, ENSMM, Besançon

Mme PRELLE Christine

Professeur, UTC, Compiègne

AUTRES MEMBRES DU JURY :

Mr BOYER Frédéric

Professeur, IMT Atlantique, Nantes

Mr CHIKHAOUI M. Taha

Chargé de recherche, CNRS, Grenoble

Avertissement au lecteur

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition des membres de la communauté universitaire. Il est soumis à la propriété intellectuelle de l'auteur. Cela implique une obligation de citation et de référencement lors de l'utilisation de ce document. D'autre part, toute contrefaçon, plagiat, reproduction ou représentation illicite encourt une poursuite pénale.

Code de la Propriété Intellectuelle

Article L122-4 :

Toute représentation ou reproduction intégrale ou partielle faite sans le consentement de l'auteur ou de ses ayants droit ou ayants cause est illicite. Il en est de même pour la traduction, l'adaptation ou la transformation, l'arrangement ou la reproduction par un art ou un procédé quelconque.

Articles L335-1 à L335-9 :

Dispositions pénales.

Licence attribuée par l'auteur



Remerciements

Je tiens à remercier particulièrement mon encadrant de thèse, Benoît Rosa. Toujours disponible et volontaire, tu as su m'accompagner et me guider avec une grande justesse tout au long de ce travail. Tu ne m'as jamais imposé tes idées, mais tu les as largement enrichies par ton expertise et tes réflexions. Nous n'avons pas seulement "parlé Sciences", tu m'as également aidé à esquisser mes aspirations professionnelles en me partageant généreusement ton expérience. Ce fut un privilège et une chance de travailler avec toi.

Je souhaite également remercier chaleureusement mon directeur de thèse, Bernard Bayle. Dès le premier jour, tu m'as accordé ta confiance, me permettant de travailler avec une autonomie que j'ai profondément appréciée. Tu as su m'accompagner avec bienveillance, tout en me rappelant les échéances importantes lorsque cela s'avérait nécessaire. Ce fut un réel plaisir de travailler sous ta direction, et ta gestion de projet restera pour moi un exemple à suivre.

Je tiens à adresser mes plus sincères remerciements à mon second directeur de thèse, Florent Nageotte. Ta rigueur scientifique exceptionnelle m'a continuellement encouragé à approfondir les concepts étudiés. Sous ta direction, j'ai appris à structurer et argumenter mes travaux avec précision et exigence, ce qui a considérablement élevé la qualité des résultats obtenus tout au long de ce projet. Je conserverai précieusement tes conseils et continuerai à m'en inspirer dans mes travaux futurs.

Merci à vous trois, Benoît, Bernard et Florent, de m'avoir formé et accompagné tout au long de ces trois années. Vous m'avez fait grandir et aimer le métier de chercheur. Je vous en serai éternellement reconnaissant, et ce sera toujours un plaisir de collaborer et d'échanger avec vous. Je vous souhaite le meilleur pour la suite.

Je tiens à remercier les membres de mon jury de thèse, Guillaume Laurent, Christine Prelle, Frédéric Boyer et M. Taha Chikhaoui, pour avoir consacré du temps à l'étude et à l'évaluation de mon travail. Ce fut un privilège de bénéficier de vos expertises respectives et d'avoir pu échanger avec vous sur les perspectives des travaux de recherches présentés.

Je tiens également à remercier Chloé Lindingre. En tant que stagiaire, tu as réalisé un travail remarquable, qui a eu un impact positif et significatif sur le déroulement de ma thèse. En tant que collègue et amie, ce fut un réel plaisir de partager avec toi mes galères, mes réussites et mes questionnements. Je t'adresse tout mon soutien et mes encouragements pour la thèse que tu prépares.

Je tiens à exprimer ma reconnaissance à Philippe Zanne, ingénieur de recherche. Ton travail sur la programmation du prototype de robot à tubes concentriques a été essentiel pour me permettre de tester expérimentalement mes contributions théoriques.

Je souhaite remercier chaleureusement Benoît Wach, ingénieur de recherche de l'équipe et ami, de m'avoir partagé ton savoir faire autour de la conception mécanique ainsi que pour tout le temps passé à la fabrication des pièces nécessaires pour le prototype de robot à tubes concentriques. Attentionné et volontaire, tu as toujours répondu présent pour me conseiller et m'aider.

Je tiens à remercier Thibault Poignonec pour tes conseils avisés lors du développement d'un algorithme de commande de robot et d'une solution de quantification probabiliste de l'erreur d'estimation des forces.

Je souhaite également exprimer ma gratitude à Manuel Yguel, ingénieur de recherche. Tu as pris de ton temps pour me conseiller dans la mise en œuvre d'une solution d'estimation de la forme par exploitation d'images du robot à tubes concentriques.

Merci à vous cinq, Chloé, Philippe, Benoît, Thibault et Manuel, pour vos contributions respectives à ce travail de thèse, qui est, d'une certaine manière, aussi un peu le vôtre. Ce fut une chance de travailler avec vous et je vous souhaite également le meilleur pour la suite.

Je tiens à remercier Pierre Renaud pour la confiance que m'as accordé et pour m'avoir permis de réaliser des missions d'enseignements à l'INSA, mon ancienne école. Cela sera toujours un plaisir de travailler et d'échanger avec toi.

Je souhaite également remercier tout particulièrement Lennart Rubbert. Avant même le début de cette thèse, tu as été mon encadrant de projet de fin d'études, lors de ma formation d'ingénieur en mécatronique et tu m'as fait découvrir, de la meilleure manière, le domaine de la recherche.

Merci à vous deux, Pierre et Lennart. Ce fut une chance d'avoir été l'un de vos élèves puis l'un de vos collègues. Il est évident que si je ne vous avais pas rencontré, je ne me serais peut-être jamais intéressé à la recherche, discipline dans laquelle que me suis épanoui durant ces trois dernières années. Je vous souhaite le meilleur pour la suite.

Je tiens à exprimer ma gratitude à tous les membres de l'équipe RDH : Laurent, Paolo, Maciej, Azad, Jonathan, Élodie et Hassan. Merci de m'avoir accueilli dans votre équipe. Votre bienveillance, votre disponibilité et vos échanges ont grandement contribué à rendre cette thèse aussi enrichissante qu'agréable.

Ce fut un réel plaisir de partager mon quotidien avec des doctorants aussi sympathiques et inspirants que vous : Loic, Timothée, Thomas, Jérémy, Fadi, Ounay, Anne, Aude, Mahdi, Guilherme, John, Gaelle et Karine. Merci pour votre bonne humeur, vos conseils et les moments partagés. Je vous souhaite à tous beaucoup de succès pour la suite de vos projets, personnels et professionnels.

Je tiens également à remercier l'équipe de l'INSA que j'ai côtoyée : Baptiste, Cléa, Vincent, Olivier, Cédric, Bastien, Renaud, David Sébastien P. et Sébastien D. Merci pour tous les bons moments passés avec vous.

Je souhaite adresser mes plus sincères remerciements à ma famille pour votre soutien inconditionnel tout au long de cette aventure, même si mon travail a pu parfois vous sembler abstrait. Vous m'avez offert un cadre rassurant et bienveillant qui m'a permis d'avancer avec sérénité.

Je tiens à remercier tout particulièrement Ophélie Evieux. Tu as été un soutien précieux et constant tout au long de cette thèse : tu t'es intéressée à mon travail, tu m'as écouté et conseillé dans les moments de doute, et tu as même participé à la relecture de mon manuscrit. Tu as vécu cette aventure à mes côtés, et il est évident que sans toi, cette expérience aurait été bien différente.

Table des matières

Table des figures	IX
Table des tableaux	XV
1 Introduction	1
1.1 Introduction à la robotique continue chirurgicale	1
1.1.1 L’histoire des robots manipulateurs	1
1.1.2 Les robots chirurgicaux	3
1.1.3 Les robots continus	5
1.2 Introduction à la modélisation de robots	8
1.2.1 Modèles de robot	8
1.2.2 Construire un modèle de robot	11
1.2.3 Construire un modèle analytique de robot	13
1.2.4 Utilisation des modèles	14
1.3 Contributions et organisation du manuscrit	15
1.4 Communications scientifiques	16
1.5 Financements	16
2 Modélisation cinétostatique directe des robots continus	19
2.1 Généralités sur les robots continus	19
2.1.1 Géométrie différentielle d’un robot continu	20
2.1.2 Actionnement et interactions avec l’environnement du robot : sources de déformation	22
2.1.3 Modélisation des déformations du robot	23
2.1.4 Modélisation de la propagation des efforts internes le long du robot : les équations de Cosserat	25
2.1.5 Construction d’un vecteur d’état et résolution du modèle différentiel	27
2.2 Cas d’étude : Modélisation cinétostatique d’un robot à tubes concentriques	31
2.2.1 Abscisse curviligne et translation des tubes	31
2.2.2 Courbures et rotation des tubes	33

2.2.3	Interaction entre les tubes	36
2.2.4	Des hypothèses au modèle différentiel	38
2.2.5	Conditions aux limites et résolution	40
2.2.6	Implémentation du modèle cinétostatique	43
2.3	Conclusions	45
3	Modélisation cinématique et compliance	47
3.1	Définition de la compliance	48
3.1.1	Introduction	48
3.1.2	La matrice de compliance généralisée	50
3.2	Calcul de la compliance généralisée	52
3.2.1	Calcul par accroissements finis numériques	52
3.2.2	Calcul hybride : littéral et différences finies numériques	53
3.2.3	Calcul littéral par propagation de dérivées haut-niveau	56
3.2.4	Calcul littéral par propagation de dérivées bas-niveau	59
3.3	Cas d'étude : Modéliser et calculer la compliance généralisée des robots à tubes concentriques	64
3.3.1	Propagation des dérivées bas-niveau	64
3.3.2	Calcul des dérivées haut-niveau	67
3.3.3	Validation numérique	68
3.3.4	Temps de calcul	75
3.4	Discussions et perspectives	77
3.4.1	Discussions sur la validation numérique	77
3.4.2	Perspectives sur l'étude du point de contact	77
3.4.3	Perspectives sur le cas multi-forces	78
3.4.4	Perspectives sur le modèle de compliance généralisée	78
3.4.5	Perspectives générales sur la compliance généralisée	79
4	Estimation d'efforts externes basée compliance	83
4.1	État de l'art	84
4.2	Estimation d'efforts grâce à une mesure de forme	86
4.2.1	Compliance généralisée et estimation d'efforts	86
4.2.2	Conditionnement du problème	87
4.2.3	Formulation d'un problème d'optimisation	89
4.2.4	Étude du problème force-position	89
4.2.5	Estimation de l'incertitude	92
4.3	Cas d'étude : Estimation des forces pour les robots à tubes concentriques	94
4.3.1	Validation numérique	94

4.3.2	Développement d'un prototype	100
4.3.3	Résultats expérimentaux	105
4.4	Discussions et perspectives	109
4.4.1	Discussions sur le banc expérimental	109
4.4.2	Discussions sur l'étude en simulation	109
4.4.3	Perspectives sur la formulation d'hypothèses	110
4.4.4	Perspectives sur l'incertitude de l'estimation de force	113
4.4.5	Perspectives sur le cas multi-forces	114
4.4.6	Perspectives générales sur l'estimation d'efforts grâce à une mesure de forme	114
5	Conclusions et perspectives	117
5.1	Conclusions générales	117
5.2	Perspectives générales	117
	Annexes	121
	Bibliographie	139

Table des figures

1.1	Schéma du canard mécanique, le premier automate élaboré par Jacques de Vaucanson en 1739 (a) et photographie de Raymond Goertz à côté du premier système maître-esclave conçu en 1954 pour manipuler des objets à distance (b). Adapté de [Scientific American, 1899] et [Sheridan, 1989]. . .	2
1.2	Photographies du robot Unimate, le premier robot manipulateur industriel créé en 1961 par George Devol (a) et du robot collaboratif UR30 commercialisé depuis 2024 par Universal Robots (b). Adapté de [Robot Magazine, 2023] et [Universal Robots, 2024]	3
1.3	Photographies de Yin San Kwoh à côté du robot PUMA pour assister une biopsie de cerveau (a) et du robot Zeus utilisé en 2001 par Jacques Marescaux pour réaliser une chirurgie transatlantique (b). Adapté de [Veneziano et al., 2020] et [Eto and Naito, 2005]	4
1.4	Photographies du robot Cirq (Brainlab, Munich, Allemagne) utilisé pour assister des biopsies de cerveau (a) et du robot Da Vinci (Intuitive Surgical, Sunnyvale, Californie, États-Unis) utilisé pour réaliser des coelioscopies minimalement invasives (b). Adapté de [Brainlab, 2024] et [Intuitive Surgical, 2024]	5
1.5	Photographies du robot hyper-redondant Tensor Arm développé en 1967 par Victor Anderson et Ronald Horn (a) et du robot doté d'un actionnement fluide développé par James Wilson en 1985 (b). Adapté de [Open Continuum Robotics, 2022b] et [Open Continuum Robotics, 2022a].	6
1.6	Schéma présentant les différentes architectures mécaniques de robots manipulateurs : les robots conventionnels ($ddl \leq 6$) (a), les robots redondants ($ddl \geq 6$) ou hyper-redondants ($ddl \gg 6$) (b) et les robots continus ($ddl \approx \infty$) (c). Schéma inspiré de [Suzumori et al., 2023].	7
1.7	Photographies de plusieurs types de robots continus : un robot à câble (a), un robot agoniste-antagoniste (b), un robot actionné grâce à un champ magnétique (c) et un robot à tubes concentriques (d). Adapté de [University of Toronto - CRL, 2021], [Oliver-Butler et al., 2021], [Chun et al., 2010] et [Mahoney et al., 2019].	7
1.8	Schéma d'un mécanisme bielle-manivelle qui permet de déplacer un effecteur grâce à la mise en rotation d'une articulation actionnée à la base. . . .	8

1.9	Paramètres géométriques du mécanisme bielle-manivelle et illustration des modèles géométriques directs et inverses.	9
1.10	Paramètres cinématiques du mécanisme bielle-manivelle et illustration des modèles cinématiques directs et inverses.	9
1.11	Paramètres dynamiques du mécanisme bielle-manivelle et illustration des modèles dynamiques directs et inverses.	10
1.12	Paramètres géométriques du mécanisme bielle-manivelle complété par le paramétrage d'une force appliquée sur l'effecteur et illustration des modèles cinéostatiques directs et inverses.	10
1.13	Modélisation géométrique directe analytique du mécanisme bielle-manivelle.	11
1.14	Représentation d'une modélisation géométrique directe du mécanisme bielle-manivelle en utilisant une méthode expérimentale.	12
2.1	Schéma illustrant les définitions de la variation de position $v_0(s)$ ainsi que de la courbure $u_0(s)$, utilisées pour décrire la forme d'un robot continu. . .	21
2.2	Illustration d'un robot continu où sont représentés les efforts externes $w_0(s_0)$ ainsi que les efforts internes $m_0(s)$ et $n_0(s)$	23
2.3	Illustration des phénomènes d'élongation (a), de compression (b), de cisaillement (c) et des déformations angulaires de flexion (d) et de torsion (e).	24
2.4	Représentation des efforts appliqués à un segment infiniment petit.	25
2.5	Illustration de la résolution d'un PVI (a) et d'un PVL (b).	29
2.6	Paramétrage des abscisses curvilignes particulières pour un RTC composé de 3 tubes.	33
2.7	Représentation du paramétrage géométrique des tubes.	34
2.8	Représentation des repères flottants et des angles de torsion pour un RTC composé de 2 tubes.	36
2.9	Représentation des conditions aux limites connues et inconnues dans le cas d'un RTC composé de 3 tubes.	41
2.10	Visualisation de la forme obtenue pour la configuration RTC1-C1 à l'issue de la résolution du modèle cinéostatique. Deux scénarios ont été simulés : lorsque le RTC ne subit aucune force d'interaction (a) et lorsque ce dernier subit les forces $\ f_0(s_0 = 30)\ = 60$ N, $\ f_0(s_0 = 80)\ = 62$ N et $\ f_0(L_0 = 100)\ = 34$ N (b).	44
3.1	Illustration et représentation matricielle de la définition de la jacobienne articulaire selon (3.2).	49
3.2	Illustration et représentation de la matrice de compliance selon (3.3).	50
3.3	Illustration graphique du concept de compliance généralisée proposé dans cette thèse (a) et de sa représentation matricielle (b).	51

3.4	Illustration de la réduction des intervalles d'intégration des dérivées partielles en fonction des points d'application des efforts externes.	61
3.5	Représentation des différentes variations regroupées dans $\Delta\Omega_{ctrl}$	69
3.6	Illustration des distances définies par (3.48) et mesurées entre la forme initiale $p^*(s)$ (a), la forme déformée calculée grâce au modèle cinéto-statique $p_{mod}(s)$ (b) et la forme déformée calculée par linéarisation grâce à la matrice de compliance généralisée et la jacobienne $p_{jacob}(s)$ (c).	71
3.7	Présentation d'un extrait de 500 formes $p^*(s)$ (sur un total de 5000 formes) simulées pour le cas sans précharge (a)-(b) et dans le cas avec précharge (c)-(d). L'extrait de 10 formes $p^*(s)$ pour le cas avec précharge (e)-(f) souligne la grande diversité des formes simulées incluant notamment des formes en "C" et en "S".	73
3.8	Déformations relatives mesurées à partir des formes simulées grâce au modèle (\mathcal{D}_{mod}^*) et grâce aux matrices $C_{s_0}(s)$ et $J(s)$ calculées par la MPDBN (\mathcal{D}_{jacob}^*). La différence entre ces deux déformations est mesurée par $\mathcal{D}_{mod}^{jacob}$. Les résultats présentés ont été obtenus à partir de 5000 simulations de formes sans précharge (a) et avec précharge (b).	74
3.9	Évolution du temps nécessaire au calcul de la compliance $C_{s_0}(s)$ en fonction du nombre de points de discrétisation N_P et pour les 4 méthodes présentées (a) et du gain de temps permis par la MPDBN par rapport à la méthode la plus efficace de la littérature, la MPDHN (b).	76
3.10	(a) Visualisation des erreurs moyennes de linéarisation $\mathcal{D}_{mod}^{jacob}$ des déformations du RTC2 en fonction du point d'application s_0 des forces et de leurs intensités $\ \Delta f_0(s_0)\ $. Les graphiques (b) et (c) illustrent respectivement les formes obtenues lorsque des forces de 100 N sont appliquées en $s_0 = \frac{L_0}{2}$ et $s_0 = \frac{3L_0}{4}$. On y retrouve la forme initiale $p^*(s)$ (d), la forme déformée estimée grâce au modèle de forme $p_{mod}(s)$ (e) et la forme déformée estimée par linéarisation grâce à la matrice de compliance généralisée $p_{jacob}(s)$ (f).	80
3.11	(a) Visualisation des erreurs moyennes de linéarisation $\mathcal{D}_{mod}^{jacob}$ des déformations du RTC2 en fonction du nombre de forces appliquées simultanément N_f et de leurs intensités $\ \Delta f_0(s_0)\ $. Les graphiques (b) et (c) illustrent respectivement les formes obtenues lorsque 5 forces de 10 N et 100 N sont appliquées simultanément sur le RTC. On y retrouve la forme initiale $p^*(s)$ (d), la forme déformée estimée grâce au modèle de forme $p_{mod}(s)$ (e) et la forme déformée estimée par linéarisation grâce à la matrice de compliance généralisée $p_{jacob}(s)$ (f).	81
4.1	Vue en 3D (a) et vue de dessus (b) de la cartographie du conditionnement de la matrice de compliance généralisée $C_{s_0}(s)$ en fonction de s et s_0 , réalisée pour la configuration RTC1-C1.	88
4.2	Illustration pour une poutre encastree-libre du phénomène de bras de levier lorsqu'une force est appliquée à sa base (a) ou à son extrémité libre (b) ainsi que de son champ de déformation (c).	88

4.3	Illustration de la variance de l'estimation de la force $\Delta\tilde{f}_0(\tilde{s}_0)$ qui peut être calculée sous condition de pouvoir quantifier la variance de toutes les déformations $\Delta p_0(s)$	93
4.4	Résultats obtenus grâce à l'algorithme d'estimation de force en simulation pour la configuration RTC3-C1. On y retrouve la force simulée $\Delta f_0(s_0)$, la force estimée $\Delta\tilde{f}_0(\tilde{s}_0)$, ainsi que : (a) la forme initiale $p^*(s)$, (b) une simulation de la forme déformée et mesurée $p_{mes}(s)$, (c) la forme déformée $p_{mod}(s)$ estimée par le modèle du RTC et (d) la forme déformée $p_{jacob}(s)$ estimée par linéarisation grâce à la matrice compliance généralisée.	96
4.5	Évolution de la fonction de coût simulée du problème d'optimisation (4.17a) en fonction du potentiel point d'application de la force s_0 pour la configuration RTC3-C1. Les minimums des fonctions de coût associées aux estimations 2D et 3D (a) sont atteints pour $\tilde{s}_0 = 140$ mm.	97
4.6	Image générée de la conception du prototype de RTC (a) et photographie du prototype réellement fabriqué (b).	101
4.7	Photographies du banc de test comprenant un poste informatique, une armoire électrique, le prototype de RTC ainsi qu'un système d'estimation de la forme du robot.	103
4.8	Images du RTC, dans la configuration RTC4-C1, prises par les deux caméras. La détection du robot est affichée en rouge sur les images.	104
4.9	Illustration de l'algorithme de reconstruction épipolaire de la forme 3D du RTC, inspiré de [Franco and Boyer, 2005], qui a été implémenté pour le banc de test expérimental.	104
4.10	Photographies superposées du RTC, dans la configuration RTC4-C1, à l'état initial (a) et lorsqu'il est déformé par une force $\ \Delta f_0(s_0 = 110)\ = 1.5$ N.	105
4.11	Présentation de la force 3D estimée ainsi que des différentes formes estimées : l'estimation brute (a) et sa représentation sous forme de spline (b) de la forme initiale, la forme initiale prédite par le modèle $p^*(s)$ (c), l'estimation brute (d) et sa représentation sous forme de spline (e) de la forme déformée, la forme déformée estimée grâce au modèle $p_{mod}(s)$ (f) et la forme déformée estimée grâce à la compliance généralisée $p_{jacob}(s)$ (g).	106
4.12	Évolution de la fonction de coût du problème d'optimisation (4.17a) en fonction du potentiel point d'application de la force s_0 obtenue lors de l'essai expérimental réalisé sur la configuration RTC4-C1. Les minimums des fonctions de coût associées aux estimations 3D (a) et 2D (c) sont atteints pour $\tilde{s}_0 = 59$. Les estimations 3D (b) et 2D (d) pour les vecteurs forces $\Delta\tilde{f}_0(s_0)$ sont obtenues en injectant l'information du point d'interaction réel s_0 dans le problème d'optimisation (4.17a).	107

4.13	Cartographies des erreurs, mesurées lors des essais en simulation de la méthode d'estimation de force, en fonction du point d'application de la force $\frac{s_0}{L_0}$ et de son intensité $\ \Delta f_0(s_0)\ $. On y retrouve les erreurs mesurées pour les estimations 2D (a)-(c) ainsi que pour les estimations 3D (d)-(f). Pour des raisons de visualisation, les orientations des graphiques ne sont pas toutes identiques : les abscisses $\frac{s_0}{L_0}$ croissantes sont indiquées par des flèches vertes et les intensités $\ \Delta f_0(s_0)\ $ croissantes par des flèches rouges.	111
4.14	Estimation et cône de probabilité de la force appliquée en simulation sur le RTC3. On y retrouve la force simulée $\Delta f_0(s_0)$, la force estimée $\Delta \tilde{f}_0(\tilde{s}_0)$, ainsi que : la forme initiale $p^*(s)$ (a), la forme déformée et mesurée $p_{mes}(s)$ (b), la forme déformée $p_{mod}(s)$ estimée par le modèle du RTC (c) et la forme déformée $p_{jacob}(s)$ estimée par linéarisation grâce à la matrice compliance généralisée (d).	112
4.15	Évolution de l'angle entre la tangente du robot $\vec{z}_0^{\rightarrow}(s)$ et la direction de moindre déformation $\vec{e}_{\ominus}^{\rightarrow}(s)$ le long des configurations RTC1-C3 (a) et RTC1-C4 (b).	112
5.1	Synthèse de la résolution du modèle cinéto-statique direct dans le cas d'un RTC	126
5.2	Synthèse de la MPDBN dans le cas d'un RTC. L'indication \bullet^* précise ici que le dérivateur χ est considéré pour $\forall i \in \llbracket 1, nbT \rrbracket$ et pour $\forall s_0 \in [0, L_0]$. Les étapes indiquées par $\bullet^{(1)}$ ne sont nécessaires que pour le calcul de la jacobienne articulaire $J(s)$ et $\bullet^{(2)}$ que pour le calcul de la matrice de compliance généralisée $C_{s_0}(s)$	127
5.3	Comparaison graphique des coefficients de la matrice de compliance généralisée $C_{s_0=80}(s)$ pour le RTC1-C2. Les courbes de référence, issues de la MDFPVL, sont indiquées en orange et celles obtenues par la MPDBN en bleu.	128
5.4	Déformations relatives mesurées à partir des formes simulées grâce au modèle (\mathcal{D}_*^{mod}) et grâce aux matrices $C_{s_0}(s)$ et $J(s)$ obtenues grâce à la MPDBN (\mathcal{D}_*^{jacob}) . La différence entre ces deux déformations est mesurée par $(\mathcal{D}_{mod}^{jacob})$. Les résultats présentés ont été obtenus à partir de 5000 simulations de formes sans précharge (a) et avec précharge (b). Contrairement à la validation numérique présentée dans la sous-section 3.3.3 où les variations $\Delta\theta_{ci}$, $\Delta\beta_{ci}$, $\Delta\tau_0(s_0)$, $\Delta f_0(s_0)$, $\Delta\tau_0(L_0)$ et $\Delta f_0(L_0)$ sont appliquées simultanément, les variations sont ici appliquées indépendamment pour trouver des variations limites générant des déformations d'amplitude semblable. . .	129

Table des tableaux

3.1	Tableau regroupant des valeurs de N_P , Y , B et U pour différents modèles de robots continus présentés dans la littérature. Les références ne précisant pas le nombre de points de discrétisation sont indiquées par la mention \bullet^* et la valeur par défaut $N_P = 100$ est alors utilisée.	62
3.2	Tableau regroupant les estimations du nombre d'opérations pour les exemples référencés dans le tableau 3.1 ainsi que les gains d'opérations permis par la méthode MPDBN proposée, relativement à la méthode MDFPVL, à la MDFPVI et à la MPDHN.	63
3.3	Synthèse des valeurs initiales des dérivées impliquées dans le calcul de $\frac{\partial y}{\partial x}(s)$	66
3.4	Synthèse des valeurs particulières des dérivées impliquées dans le calcul de $\frac{\partial y}{\partial x}(s)$	66
3.5	Synthèse des variations maximales appliquées sur la forme initiale $p^*(s)$ lors de la validation, pour la configuration RTC2-C1. Ces variations sont à l'origine des formes déformées $p_{mod}(s)$ et $p_{jacob}(s)$	71
3.6	Synthèse des précharges limites appliquées permettant d'obtenir des formes initiales $p^*(s)$ déformées pour les scénarios avec précharge.	72
3.7	Mesures statistiques extraites à partir des formes simulées grâce au modèle (\mathcal{D}_{mod}^*) et grâce aux matrices $C_{s_0}(s)$ et $J(s)$ calculées par la MPDBN (\mathcal{D}_{jacob}^*). La différence entre ces deux déformations est mesurée par $\mathcal{D}_{mod}^{jacob}$. Les résultats présentés ont été obtenus à partir de 5000 simulations pour des formes sans précharge (a) et avec précharge (b).	74
4.1	Résultats des estimations 2D et 3D de forces obtenues en simulation pour la configuration RTC3-C1.	98
4.2	Résultats statistiques des estimations de forces 2D et 3D obtenues en simulation pour les 16000 forces simulées sur la configuration RTC3-C1. <i>min</i> indique la valeur minimale, <i>med</i> la valeur médiane, <i>IQR</i> la valeur de l'écart inter-quartiles et <i>max</i> la valeur maximale.	99
4.3	Caractéristiques géométriques des sections des tubes du RTC.	100

4.4	Résultats des trois tests d'estimation de forces obtenus expérimentalement pour la configuration RTC4-C1 pour des points d'interaction situés à 70 mm, 100 mm et 150 mm le long du RTC. Le banc expérimental développé ne permettant pas de mesurer la direction réelle de la force appliquée sur le robot, les erreurs $\alpha(\Delta f_0(s_0), \Delta \tilde{f}_0(\tilde{s}_0))$ n'ont donc pas été quantifiées.	108
5.1	Caractéristiques et configurations du RTC1	122
5.2	Caractéristiques et configurations du RTC2	123
5.3	Caractéristiques et configurations du RTC3	124
5.4	Caractéristiques et configurations du RTC4	125
5.5	Listes des abréviations	130
5.6	Nomenclature des opérateurs mathématiques	131
5.7	Nomenclature des symboles périphériques aux variables	132
5.8	Nomenclature des variables associées au modèle cinétostatique des robots continus 1/2	133
5.9	Nomenclature des variables associées au modèle cinétostatique des robots continus 2/2	134
5.10	Nomenclature des variables associées aux robots à tubes concentriques	135
5.11	Nomenclature des variables associées au modèle de compliance des robots continus	136
5.12	Nomenclature des variables associées à l'estimation d'efforts	137

Chapitre 1

Introduction

Ce premier chapitre a pour but de présenter le contexte des travaux de recherche développés dans cette thèse. La [section 1.1](#) introduit le concept de *robot continu*, notamment dans le domaine de la chirurgie. La [section 1.2](#) présente la notion de *modèle* de robot ainsi que les méthodes couramment utilisées pour les construire. Les contributions scientifiques de nos travaux et l’organisation du manuscrit seront présentées à la [section 1.3](#), et les communications scientifiques réalisées listées à la [section 1.4](#).

Les abréviations introduites dans ce chapitre sont disponibles en annexes dans le tableau [5.5](#).

1.1 Introduction à la robotique continue chirurgicale

Cette section a pour but d’introduire les *robots continus* ainsi que leurs applications médicales, notamment en chirurgie. La [sous-section 1.1.1](#) présente tout d’abord un aperçu chronologique de l’émergence de la robotique industrielle dans nos sociétés. La [sous-section 1.1.2](#) explique ensuite comment les robots industriels ont évolué pour assister les chirurgiens dans les blocs opératoires. Enfin, la [sous-section 1.1.3](#) présente une branche de la robotique, la *robotique continue*, ainsi que son potentiel dans le contexte des *chirurgies minimalement invasives*. Ce travail bibliographique s’appuie notamment sur des articles de synthèse [[Troccaz, 2012](#)], [[Burgner-Kahrs et al., 2015](#)], [[Dupont et al., 2022](#)], [[Russo et al., 2023](#)], ainsi que sur des ressources disponibles en ligne [[Stan Institute, 2018](#)], [[Burgner-Kahrs, 2022](#)].

1.1.1 L’histoire des robots manipulateurs

Depuis longtemps, les *automates* font partie du paysage technologique de nos sociétés. Les premiers mécanismes ont été développés dans un objectif de divertissement, comme ce fut le cas avec le canard mécanique, illustré par la [figure 1.1a](#), présenté par Jacques de Vaucanson en 1739, ou encore les automates créés par la famille Jacquet-Droz de 1767 à 1774. Avec les avancées scientifiques dans le domaine de l’électricité, de nouvelles fonc-

tionnalités sont apparues, comme le contrôle à distance démontré en 1898 par Nikola Tesla avec un bateau télécommandé [Thompson, 2015]. Depuis la révolution industrielle du XIX^e siècle, les automates ont gagné en complexité, évoluant de systèmes relativement simples, créés pour divertir, à des machines semi-automatiques sophistiquées capables de mécaniser des tâches industrielles.

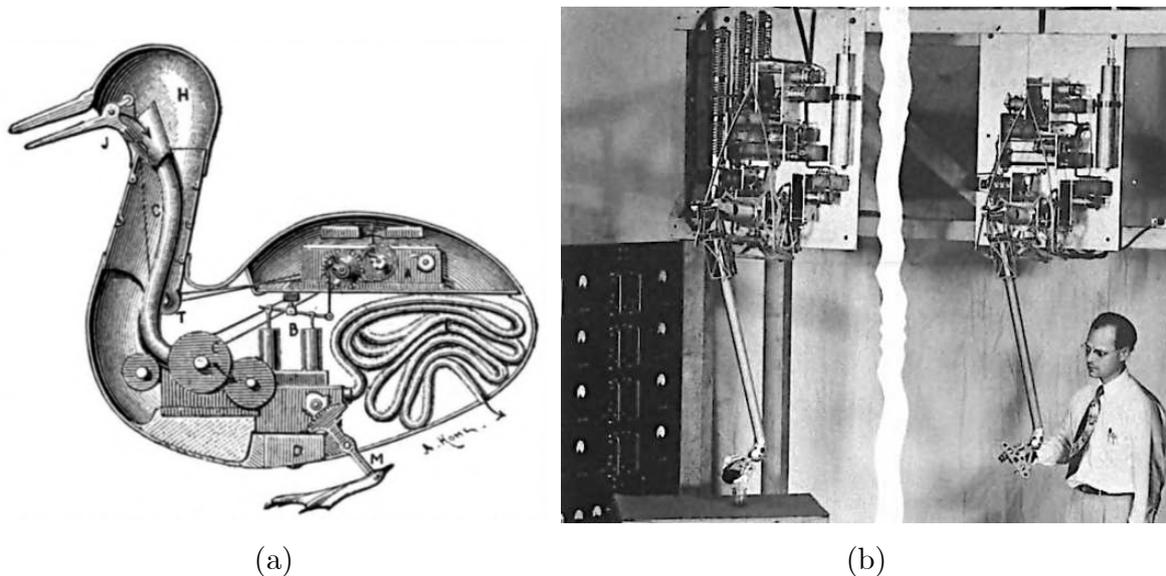


FIGURE 1.1 – Schéma du canard mécanique, le premier automate élaboré par Jacques de Vaucanson en 1739 (a) et photographie de Raymond Goertz à côté du premier système maître-esclave conçu en 1954 pour manipuler des objets à distance (b). Adapté de [Scientific American, 1899] et [Sheridan, 1989].

C'est en 1920 que le terme *robot* a été introduit par l'écrivain tchèque Karel Čapek pour désigner des machines automatiques à l'apparence humaine [Čapek, 1920]. À partir des années 1940, avec l'essor de l'industrie nucléaire, les robots s'avèrent essentiels pour manipuler des objets dangereux à distance. C'est en 1949 que l'américain Raymond Goertz a présenté le premier système électromécanique de type *maître-esclave*, illustré par la figure 1.1b, qui permet de reproduire à distance les mouvements d'un bras maître sur un bras manipulateur esclave. Dans les mêmes années, George Devol a développé le premier robot industriel totalement automatique, le robot Unimate illustré par la figure 1.2a, qui a été intégré en 1961 dans les lignes de productions du constructeur automobile américain General Motors. La même année, George Devol s'est associé à Joseph Engelberger pour fonder Unimation, la première entreprise de fabrication de robots industriels, qui connaîtra un grand succès à partir de 1978 avec la commercialisation du robot industriel programmable PUMA.

Bien que les robots industriels aient été continuellement améliorés, une limite à long-temps subsisté : ces machines ne sont en effet pas adaptées pour travailler en collaboration avec des opérateurs humains. J. Edward Colgate et Michael Peshkin ont résolu cette limite

en développant le premier *robot collaboratif*, ou *cobot*, conçu pour interagir physiquement et en toute sécurité avec des humains [Colgate et al., 1996]. Les robots collaboratifs ont été grandement perfectionnés depuis et des systèmes comme le robot UR30 (Universal Robots, Odense, Danemark), illustré par la figure 1.2b, sont désormais couramment utilisés dans les industries manufacturières.

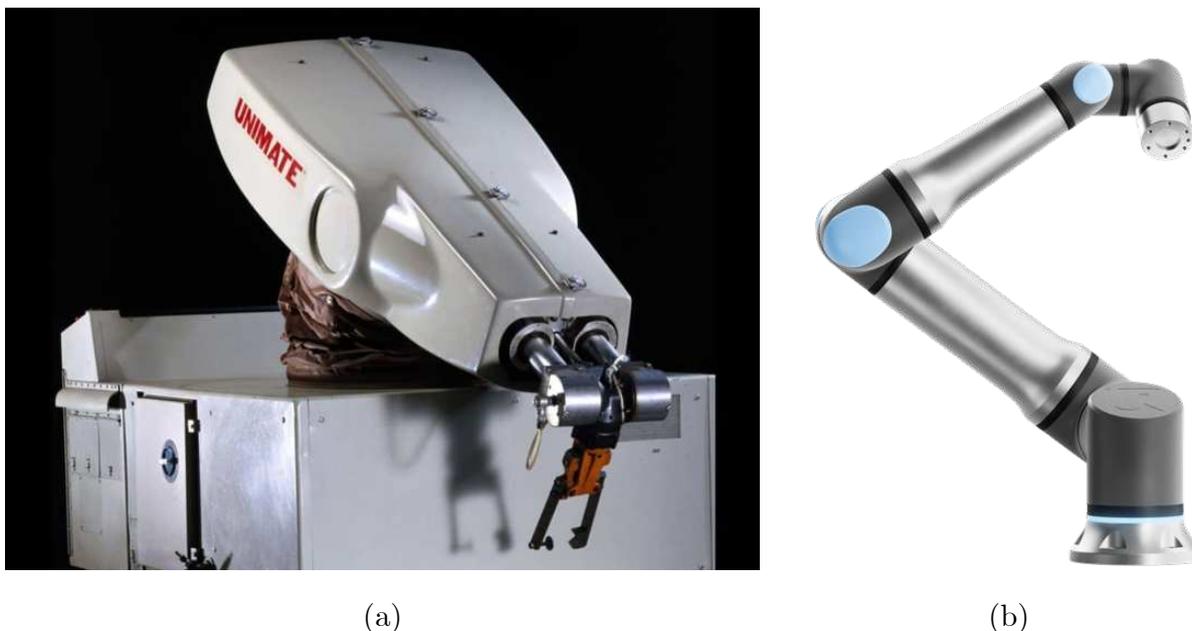


FIGURE 1.2 – Photographies du robot Unimate, le premier robot manipulateur industriel créé en 1961 par George Devol (a) et du robot collaboratif UR30 commercialisé depuis 2024 par Universal Robots (b). Adapté de [Robot Magazine, 2023] et [Universal Robots, 2024]

Historiquement, le développement des robots manipulateurs a été porté par l'industrie qui y a trouvé un intérêt économique. Encore actuellement, la robotique reste intimement liée à l'industrie et les fabricants de robots industriels ne cessent de perfectionner leurs systèmes pour pouvoir automatiser toujours plus de tâches.

1.1.2 Les robots chirurgicaux

A la suite de la démocratisation des robots dans l'industrie, d'autres secteurs ont trouvé un intérêt à utiliser des systèmes robotisés pour des applications non-manufacturières. En plus d'être infatigables, capables de manipuler des charges de façon dextre et automatique, les robots peuvent également être très précis et évoluer en contact avec des humains. Pour ces raisons, les robots ont progressivement intégré les blocs opératoires.

En 1983, James McEwen, Geof Auchinleck et Brian Day ont réalisé en 1984 la première chirurgie (une arthroscopie) assistée par le robot Arthrobot spécialement conçu pour maintenir la jambe du patient. En 1985, Yin San Kwoh détourne le robot PUMA (Unimation, USA), initialement conçu pour des tâches industrielles, pour faciliter le positionnement

d'aiguilles lors d'une biopsie de cerveau, illustrée par la [figure 1.3](#) [Kwoh et al., 1988]. Ces interventions pionnières ont montré que les systèmes robotisés peuvent améliorer la qualité des soins prodigués aux patients tout en augmentant le confort du chirurgien. D'abord utilisés comme des équipements d'assistance, Brian Davies a démontré en 1991 avec une ablation de prostate réalisée grâce au robot Probot, que les robots peuvent interagir directement avec les patients [Davies et al., 1991]. L'année suivante, une chirurgie orthopédique a marqué un tournant dans l'utilisation des robots chirurgicaux. Alors que les robots étaient jusqu'alors contrôlés par les chirurgiens, le robot Robodoc a exécuté pour la première fois des tâches chirurgicales de manière automatique [Paul et al., 1992]. En plus d'améliorer la précision des actes chirurgicaux, les robots rendent possibles des opérations chirurgicales à distance des patients, comme cela a été démontré en 2001 par Jacques Marescaux lorsqu'il a piloté depuis New-York le robot Zeus, illustré par la [figure 1.3b](#), pour réaliser une ablation de la vésicule biliaire sur une patiente prise en charge à Strasbourg [Marescaux et al., 2001].



FIGURE 1.3 – Photographies de Yin San Kwoh à côté du robot PUMA pour assister une biopsie de cerveau (a) et du robot Zeus utilisé en 2001 par Jacques Marescaux pour réaliser une chirurgie transatlantique (b). Adapté de [Veneziano et al., 2020] et [Eto and Naito, 2005]

Depuis, de nouveaux robots chirurgicaux ont été développés pour réaliser de nouvelles interventions médicales ou chirurgicales. On peut citer entre autres le robot Sensei (Hansen Medical, Mountain View, Californie, États-Unis) apparu en 2007 pour traiter des pathologies cardiaques ou encore les robots ROSA (Zimmer Biomet Robotics, Montpellier, France) et Cirq (Brainlab, Munich, Allemagne) présenté par la [figure 1.4a](#), mis en service respectivement à partir de 2011 et 2020, pour réaliser des interventions en neurochirurgie. Actuellement, le robot le plus utilisé reste le robot Da Vinci (Intuitive Surgical, Sunnyvale, Californie, États-Unis), présenté par la [figure 1.4b](#), commercialisé depuis 2000 pour réaliser des cœlioscopies minimalement invasives.

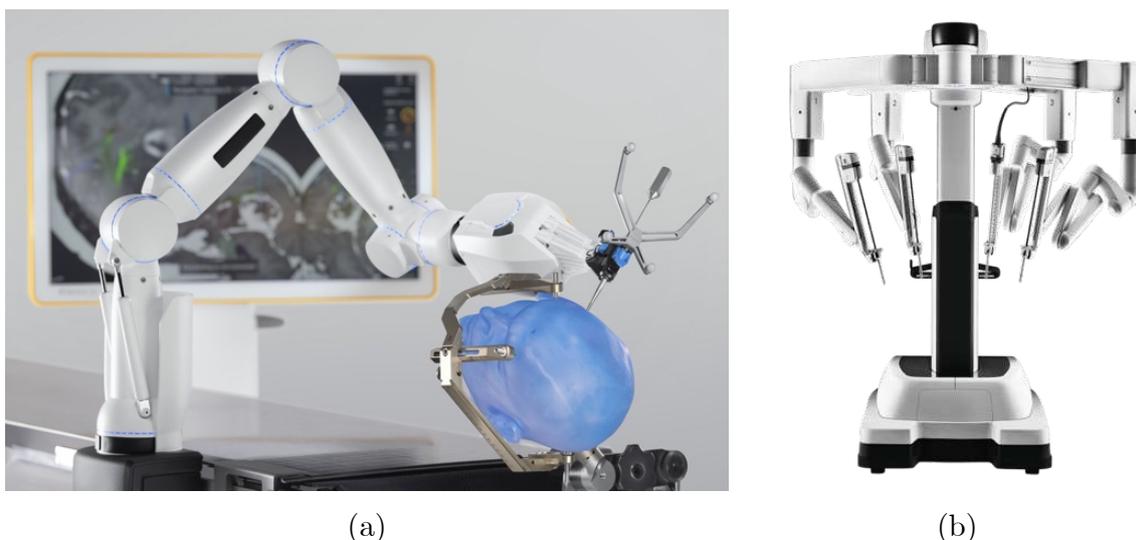


FIGURE 1.4 – Photographies du robot Cirq (Brainlab, Munich, Allemagne) utilisé pour assister des biopsies de cerveau (a) et du robot Da Vinci (Intuitive Surgical, Sunnyvale, Californie, États-Unis) utilisé pour réaliser des coéloscopies minimalement invasives (b). Adapté de [Brainlab, 2024] et [Intuitive Surgical, 2024]

Les robots actuellement utilisés en chirurgie ont notamment pour vocation de réduire le traumatisme causé au patient car ils sont associés à des *chirurgies minimalement invasives*, réduisant aussi de fait le temps de convalescence et les risques de complications. Pour permettre de telles avancées, les robots doivent pouvoir être miniaturisés tout en garantissant une grande dextérité lors de la manipulation des outils.

1.1.3 Les robots continus

Améliorer la dextérité d'un robot nécessite le plus souvent d'augmenter le nombre d'articulations du système. La mécanique définit la notion de *degré(s) de liberté (ddl)* d'un objet rigide pour représenter le nombre de mouvements indépendants possibles pour cet objet. De manière analogue, on définit le degré de liberté d'un robot comme étant le degré de liberté de son effecteur¹. Pour qu'un objet puisse être déplacé librement, il lui faut un total de 6 degrés de liberté indépendants tandis qu'un objet fixe n'en aura aucun. Afin de manipuler librement un objet ou un outil, un robot doit avoir 6 degrés de liberté au niveau de l'effecteur et doit donc intégrer 6 articulations actionnées indépendantes. Il est question de *robot redondant* par exemple lorsque la tâche implique les 6 degrés de liberté au niveau de l'effecteur mais que le nombre d'articulations actionnées est supérieur à 6 et de *robot hyper-redondant* lorsque ce nombre est très supérieur à 6. Un robot redondant offre d'intéressantes possibilités de commande : en plus de manipuler librement un objet, il peut déplacer ses liaisons sans déplacer son effecteur, fonctionnalité utile par exemple lorsque le robot évolue dans un environnement comprenant des obstacles. Ces possibilités ont été explorées dès 1967 par Victor Anderson et Ronald Horn avec le robot

1. De manière très usuelle mais moins rigoureuse, on définit par degré de liberté d'un robot le nombre total d'articulations mécaniques indépendantes et actionnées.

hyper-redondant Tensor Arm, illustré par la [figure 1.5a](#), qui est composé d'une dizaine d'articulations actionnées grâce à des câbles [[Anderson and Horn, 1970](#)].

Un tournant conceptuel important a été atteint lorsque Shigeo Hirose a présenté le robot manipulateur Elastor [[Hirose, 1983](#)]. Ce robot n'est plus constitué d'articulations discrètes conventionnelles, mais utilise des ressorts pour répartir les déformations de manière continue sur toute la longueur du robot. S'inspirant des mécanismes biologiques, Adam Morecki a développé en 1988 une trompe d'éléphant robotisée, actionnée elle aussi grâce à des câbles [[Morecki et al., 1988](#)]. De nouvelles solutions d'actionnement ont été développées au fur et à mesure des avancées technologiques comme en 1985 avec un robot à actionnement fluïdique [[Wilson, 1985](#)], illustré par la [figure 1.5b](#), et en 1988 avec un robot utilisant un actionneur réalisé à l'aide d'un alliage à mémoire de forme, dont la forme varie avec la température [[Ikuta et al., 1988](#)].



FIGURE 1.5 – Photographies du robot hyper-redondant Tensor Arm développé en 1967 par Victor Anderson et Ronald Horn (a) et du robot doté d'un actionnement fluïdique développé par James Wilson en 1985 (b). Adapté de [[Open Continuum Robotics, 2022b](#)] et [[Open Continuum Robotics, 2022a](#)].

C'est en 1999 que Graham Robinson et Bruce Davies ont introduit le terme de *robots continus* pour distinguer les robots manipulateurs sans articulation mécanique conventionnelle [[Robinson and Davies, 1999](#)]. La [figure 1.6](#) illustre les différents types de robots manipulateurs : conventionnels, redondants et continus. Depuis les premiers Robots à Câbles (RC²), d'autres solutions d'actionnement de robots continus ont été développées, comme les Robots à Tubes Concentriques (RTC³) [[Mahoney et al., 2019](#)], les Robots actionnés grâce à un champ Magnétique (RM⁴) [[Yang et al., 2023](#)], les Robots Agonistes-Antagonistes (RAA⁵) [[Tummers et al., 2024](#)] ou encore les Robots à Expansion (RE⁶) [[Blumenschein et al., 2020](#)]. Quelques types de robots continus sont illustrés par la [figure 1.7](#).

2. "Tendon-Actuated Continuum Robot" (TACR) en anglais.
3. "Concentric Tubes Continuum Robot" (CTCR) en anglais.
4. "Magnetic Continuum Robot" (M-CR) en anglais.
5. "Concentric Agonist-Antagonist Robot" (CAAR) ou "Concentric Push-Pull Robot" (CPPR) en anglais.
6. "Growing Robot" (GR) ou "Vine Robot" (VR) en anglais.

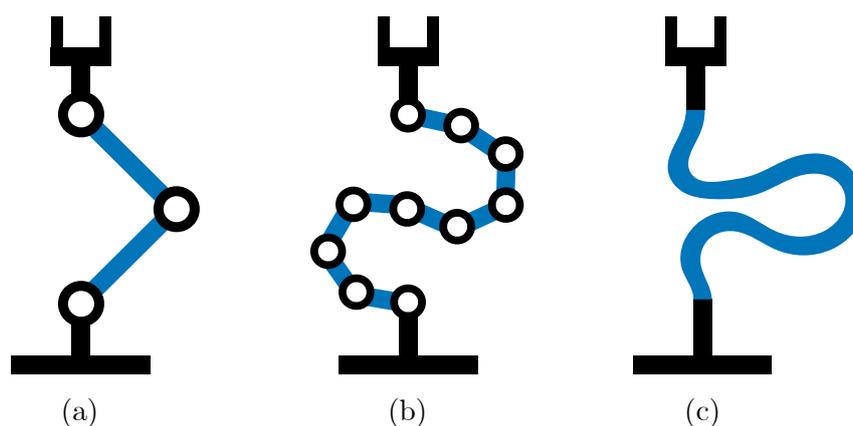


FIGURE 1.6 – Schéma présentant les différentes architectures mécaniques de robots manipulateurs : les robots conventionnels ($ddl \leq 6$) (a), les robots redondants ($ddl \geq 6$) ou hyper-redondants ($ddl \gg 6$) (b) et les robots continus ($ddl \approx \infty$) (c). Schéma inspiré de [Suzumori et al., 2023].

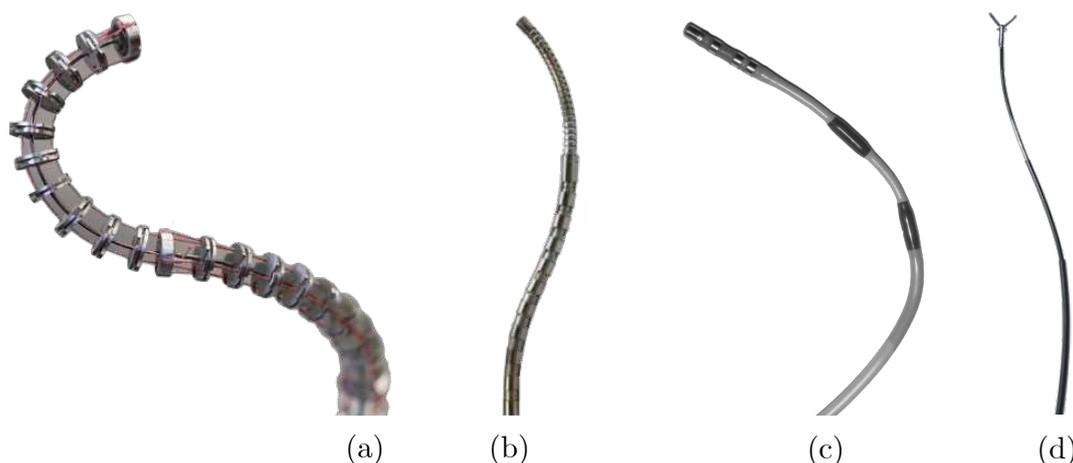


FIGURE 1.7 – Photographies de plusieurs types de robots continus : un robot à câble (a), un robot agoniste-antagoniste (b), un robot actionné grâce à un champ magnétique (c) et un robot à tubes concentriques (d). Adapté de [University of Toronto - CRL, 2021], [Oliver-Butler et al., 2021], [Chun et al., 2010] et [Mahoney et al., 2019].

Les robots continus ont un fort potentiel pour de nouvelles applications, dont les applications médicales. Alors que les robots industriels sont habituellement de grande taille et dotés d'une force importante, les robots continus sont quant à eux plus facilement miniaturisables. Par ailleurs, leur flexibilité intrinsèque les rend particulièrement adaptés à des environnements exigus et parfois fragiles. Envisagés pour des tâches d'inspection souterraine [Coad et al., 2019], spatiale [Wooten et al., 2018] ou industrielle [Dong et al., 2022], ils sont surtout particulièrement intéressants pour la chirurgie minimalement invasive [Burgner-Kahrs et al., 2015, Dupont et al., 2022]. La robotique continue est ainsi un champ de recherche très actif dans le domaine, suscitant de nouveaux travaux en modélisation, conception mécanique, commande ainsi que de nouvelles applications chirurgicales.

1.2 Introduction à la modélisation de robots

Cette section vise à introduire le concept de *modélisation de robot* qui est la plupart du temps incontournable pour commander un robot ou planifier ses mouvements. Une description des différents modèles de robot sera proposée par la [sous-section 1.2.1](#). Ensuite, plusieurs méthodes permettant de construire de tels modèles seront présentées au cours de la [sous-section 1.2.2](#) puis l'une de ces méthodes sera particulièrement développée par la [sous-section 1.2.3](#). Enfin, des exemples d'utilisation des modèles seront présentés en [sous-section 1.2.4](#). Ce travail bibliographique s'appuie également sur des articles de synthèse [Webster III and Jones, 2010], [Chikhaoui and Rosa, 2022].

Tout au long de cette section, nous illustrerons les différentes notions présentées sur un système mécanique simple, le mécanisme de bielle-manivelle, représenté en [figure 1.8](#).

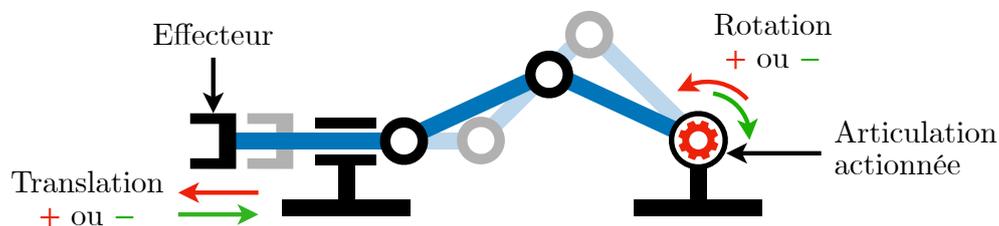


FIGURE 1.8 – Schéma d'un mécanisme bielle-manivelle qui permet de déplacer un effecteur grâce à la mise en rotation d'une articulation actionnée à la base.

1.2.1 Modèles de robot

Lorsque l'on souhaite construire un modèle, il faut tout d'abord définir les variables d'entrée et de sortie que l'on souhaite relier. Le modèle le plus élémentaire en robotique est le *modèle géométrique*. Il consiste le plus souvent à relier les variables décrivant la position des articulations à la pose de l'effecteur, c'est-à-dire à sa position et son orientation. Lorsque les variables d'entrée du modèle sont les variables articulaires, on parle de *modèle géométrique direct* et lorsque les variables d'entrée sont les variables qui décrivent la pose de l'effecteur, on parle de *modèle géométrique inverse*. Dans le cas du mécanisme bielle-manivelle présenté à la [figure 1.9](#), la variable articulaire est l'angle de l'articulation, noté α , et la position de l'effecteur est la distance notée x , tous deux mesurés par rapport à des références fixes arbitraires précisées sur la figure. Lorsque l'on dispose du modèle géométrique inverse, il devient possible de calculer l'angle α qui va placer l'effecteur à une position x souhaitée.

Les premiers modèles géométriques directs développés pour des robots continus ont utilisé l'hypothèse que les robots préservent des courbures constantes par morceaux sur toute leur longueur [Hannan and Walker, 2000], [Jones and Walker, 2006]. Cette hypothèse a ensuite été relaxée avec les modèles construits à partir des équations de Cosserat [Dupont et al., 2009], [Neumann and Burgner-Kahrs, 2016], [Rao et al., 2021].

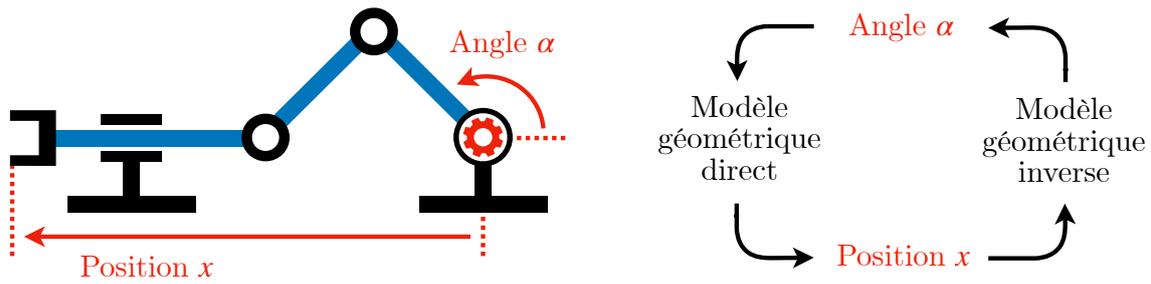


FIGURE 1.9 – Paramètres géométriques du mécanisme bielle-manivelle et illustration des modèles géométriques directs et inverses.

En complément du modèle géométrique du robot, il est également possible de construire son *modèle cinématique* destiné à modéliser sa vitesse de déplacement. Le *modèle cinématique direct* relie les vitesses articulaires à la vitesse de déplacement de l'effecteur du robot. Dans le cas du mécanisme bielle-manivelle présenté par la figure 1.10, la vitesse articulaire est la vitesse de rotation $\frac{d\alpha(t)}{dt}$, et la vitesse de l'effecteur est $\frac{dx(t)}{dt}$. Lorsque l'on dispose du *modèle cinématique inverse*, il devient possible de calculer la vitesse $\frac{d\alpha(t)}{dt}$ permettant de déplacer l'effecteur à la vitesse $\frac{dx(t)}{dt}$ souhaitée. En pratique, les modèles cinématiques de robots continus consistent le plus souvent à calculer ou estimer la Jacobienne du robot [Rucker and Webster, 2011], [Jones and Walker, 2005], [Xu et al., 2013]. Le concept de Jacobienne sera présenté en toute rigueur dans le chapitre 3.

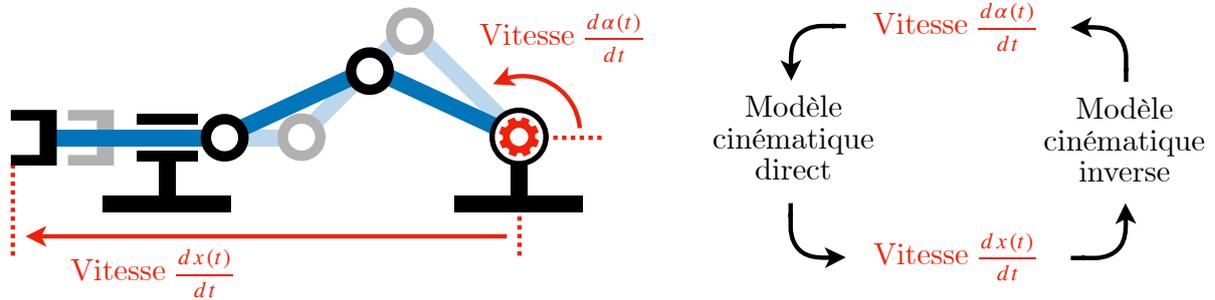


FIGURE 1.10 – Paramètres cinématiques du mécanisme bielle-manivelle et illustration des modèles cinématiques directs et inverses.

La grande majorité des robots sont pilotés à partir du seul modèle cinématique. Lorsqu'un robot se déplace particulièrement vite ou lorsqu'il subit des interactions mécaniques avec son environnement, le modèle cinématique ne permet plus de prédire fidèlement le comportement du robot. Un autre modèle peut alors être utilisé, le *modèle dynamique* qui fait intervenir l'accélération du robot ainsi que d'éventuelles forces appliquées sur ce dernier. Contrairement aux deux modèles précédents, le modèle dynamique intègre la notion de masse du système modélisé. Dans le cas du mécanisme bielle-manivelle, illustré par la figure 1.11, le *modèle dynamique direct* permet de calculer l'accélération de l'effecteur $\frac{d^2x(t)}{dt^2}$, générée par un couple d'actionnement C .

Les modèles dynamiques permettent de contrôler précisément un robot même si ce dernier se déplace très rapidement. Bien que les modèles dynamiques soient plus complexes à développer que les modèles géométriques et cinématiques, quelques modèles dynamiques ont toutefois été développés pour les robots hyper-redondants [Boyer et al., 2011], pour les robots continus en général [Till et al., 2019, Boyer et al., 2020] et pour les RTC en particulier [Renda et al., 2021].

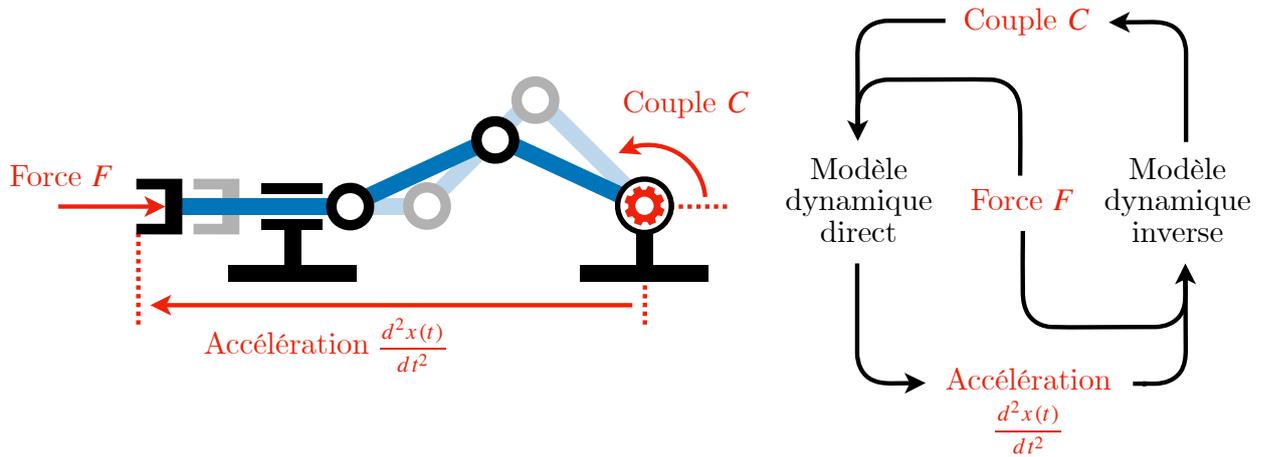


FIGURE 1.11 – Paramètres dynamiques du mécanisme bielle-manivelle et illustration des modèles dynamiques directs et inverses.

Une version simplifiée du modèle dynamique est le modèle *modèle cinétostatique* : ce modèle consiste à supposer que le robot reste statique même lorsqu'il est soumis à des forces. Cette hypothèse facilite la construction du modèle tout en y incluant les forces. Dans le cas du mécanisme bielle-manivelle, figure 1.12, le *modèle cinétostatique direct* permet de prédire la position de l'effecteur x associée à un angle de l'actionneur α , connaissant la force F appliquée sur l'effecteur.

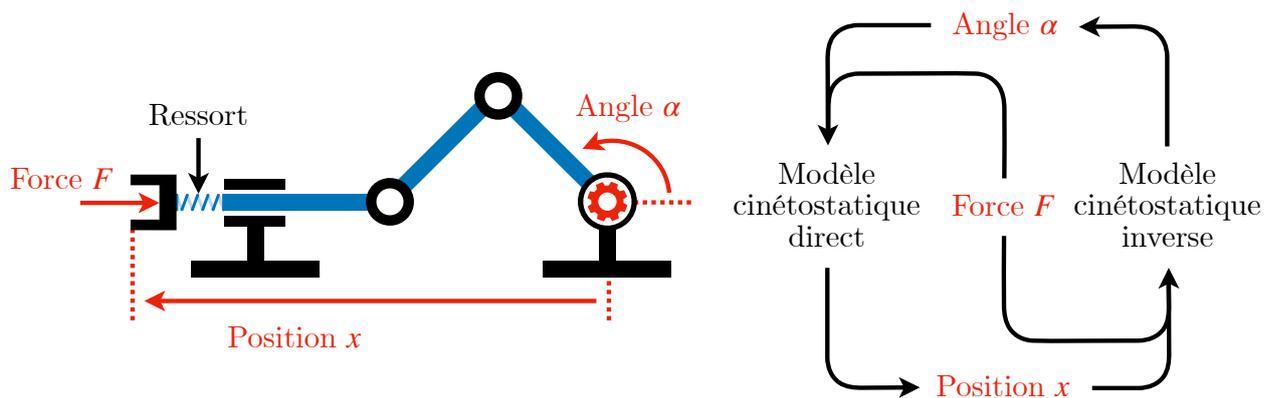


FIGURE 1.12 – Paramètres géométriques du mécanisme bielle-manivelle complété par le paramétrage d'une force appliquée sur l'effecteur et illustration des modèles cinétostatiques directs et inverses.

Les modèles cinéostatiques font certainement partie des modèles les plus utilisés actuellement en robotique continue. De nombreux modèles ont été développés pour tous les types de robots continus existants : les robots agonistes-antagonistes [Tummers et al., 2024], les robots à câbles [Oliver-Butler et al., 2019] ou encore les robots à tubes concentriques [Rucker et al., 2010], [Lock et al., 2010].

1.2.2 Construire un modèle de robot

Lorsque l'on souhaite construire un modèle, différentes méthodes peuvent être utilisées, chacune d'elle ayant ses atouts et ses inconvénients. La méthode la plus courante consiste à construire une fonction analytique et explicite qui relie les variables d'entrée et de sortie. Dans le cas du mécanisme bielle-manivelle à la figure 1.13, il est par exemple possible d'utiliser les relations géométriques pour construire le modèle géométrique direct.

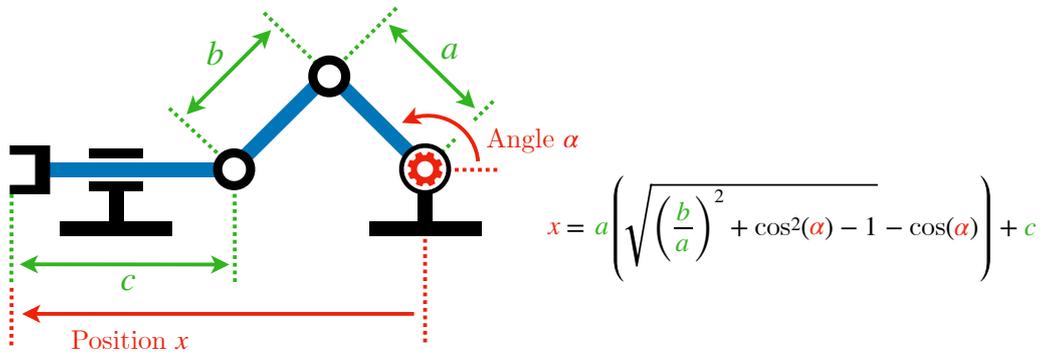


FIGURE 1.13 – Modélisation géométrique directe analytique du mécanisme bielle-manivelle.

Une modélisation analytique nécessite d'admettre des hypothèses simplificatrices ainsi qu'un domaine de validité. Dans le cas de l'exemple présenté à la figure 1.13, l'hypothèse implicite est que les segments du mécanisme sont indéformables. Dans le cas où la forme de ces segments serait modifiée, les seuls paramètres a , b et c ne permettraient alors plus de décrire convenablement la géométrie du mécanisme et il faudrait utiliser des paramètres supplémentaires. On comprend par cet exemple que les hypothèses de modélisation sont essentielles pour apprécier le niveau de réalisme d'un modèle.

Lorsque le robot continu subit des déformations importantes, il peut devenir complexe de paramétrer sa géométrie et donc de construire un modèle analytique. Dans le cas où le robot continu peut être assimilé à un système monolithique déformable, une approche alternative est alors possible : la modélisation par éléments finis. Modélisé non plus de manière analytique mais comme un ensemble de petits éléments, le robot peut être simulé pour évaluer les déplacements au niveau de chaque élément le discrétisant, et l'opération propagée de proche en proche pour finalement modéliser numériquement le robot dans sa globalité. Cette méthode, qui nécessite une puissance de calcul importante, permet de libérer le modèle d'un paramétrage particulièrement complexe mais ne produit que des résultats numériques. Bien qu'il ne soit pas possible d'obtenir de relation explicite avec cette

méthode, elle reste largement utilisée dans l'étude des structures déformables et notamment en robotique souple [Lee et al., 2017, Wu and Zheng, 2022, Cignolini et al., 2023].

Les méthodes analytiques et par éléments finis sont toutes les deux basées sur des modèles théoriques et leurs hypothèses associées qui sont parfois complexes à identifier et à adapter pour un robot continu. Dans ce cas, il est possible d'utiliser une méthode expérimentale ne nécessitant pas de modèle physique sous-jacent, et donc pas d'hypothèse simplificatrice, mais plutôt un modèle purement mathématique et paramétrique. Ce type d'approche consiste à instrumenter le robot pour mesurer les données d'entrée et de sortie. En recueillant des données pour une multitude de scénarios d'actionnement, il est alors possible d'estimer des paramètres du modèle mathématique permettant de relier les données d'entrées et les données de sortie. Un exemple de modèle expérimental est illustré par la figure 1.14.

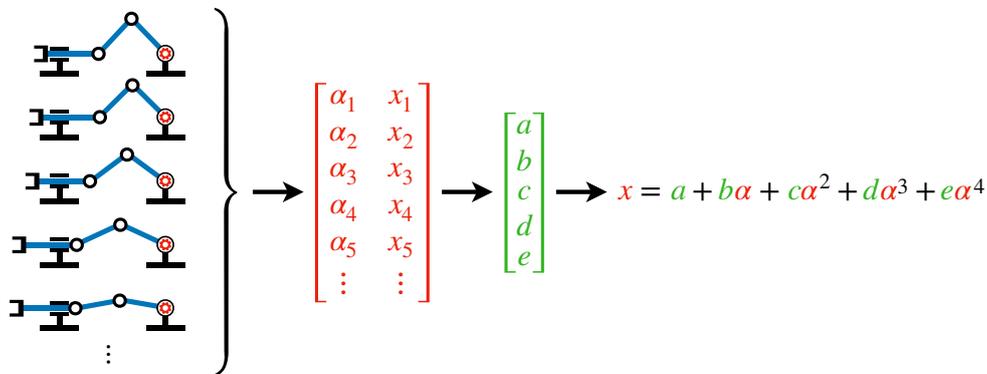


FIGURE 1.14 – Représentation d'une modélisation géométrique directe du mécanisme bielle-manivelle en utilisant une méthode expérimentale.

Les modèles expérimentaux ne sont pas exclusivement destinés à la robotique mais représentent un domaine de recherche à part entière et très actif depuis la montée en puissance de l'intelligence artificielle. Des méthodes complémentaires ont été développées pour permettre un réglage automatique des paramètres de modèles : on parle alors d'apprentissage automatique. Ce type de méthode a permis d'utiliser des gabarits mathématiques très complexes pour améliorer le réalisme des modèles. Un exemple de gabarit simple peut être une fonction de transfert du premier ordre, paramétrée par seulement 2 paramètres et très utilisée en automatique théorique. A l'opposé, le gabarit le plus complexe réalisé en 2024 serait certainement le modèle utilisé par le programme GPT-3 composé de 175 milliards de paramètres différents [Brown, 2020].

Les modèles expérimentaux peuvent être utilisés pour modéliser des phénomènes et des robots très complexes, et, lorsque le nombre de paramètres reste raisonnable, permettent d'obtenir des relations analytiques entre les entrées et les sorties. Cette méthode a notamment été utilisée pour des robots à câbles afin de construire des modèles géométriques [Xu et al., 2017] et cinématiques [Thuruthel et al., 2016, Aleluia Porto et al., 2019]. Cependant, cette approche admet une limite majeure : le réalisme du modèle est directement

dépendant des données utilisées lors de son réglage. Dans le cas où des données biaisées seraient utilisées, il devient difficile de prédire l'impact engendré sur la qualité du modèle. De manière analogue, si le modèle est utilisé pour des entrées très différentes de celles utilisées pour le réglage, le réalisme des sorties générées est inconnu.

1.2.3 Construire un modèle analytique de robot

Lorsque l'on souhaite construire un modèle analytique de robot, deux approches différentes mais équivalentes sont envisageables : les approches newtonienne et lagrangienne. En fonction de la nature du mécanisme à modéliser, une méthode peut faciliter la mise en équation du modèle analytique. L'approche newtonienne consiste à ré-utiliser le modèle fondamental de Isaac Newton qui énonce que : *"les changements qui arrivent dans le mouvement sont proportionnels à la force motrice et se font dans la ligne droite dans laquelle cette force a été imprimée"* [Newton, 1756]. Ce principe est traduit mathématiquement par le fait que tout corps ayant une quantité de mouvement \vec{p} et étant soumis à des forces \vec{F}_i est régi dans un référentiel galiléen par la loi :

$$\frac{d\vec{p}}{dt} = \sum_i \vec{F}_i \quad (1.1)$$

Cette équation met en relation les forces qui s'exercent sur un corps avec son déplacement. Cette égalité vectorielle est particulièrement adaptée à la modélisation d'un robot lorsque les forces et les déplacements peuvent être décrits dans des repères cartésiens, où ils peuvent être projetés. Cette approche a notamment été utilisée pour modéliser les robots à câbles [Camarillo et al., 2008] ainsi que les robots à tubes concentriques [Lock et al., 2010, Rucker et al., 2010]. L'effort en terme de calcul requis par cette approche s'accroît à mesure que le nombre de forces considérées augmente. Par ailleurs, lorsque les repères cartésiens ne s'avèrent pas adaptés pour décrire le mouvement d'un mécanisme, il est possible d'utiliser d'autres types de repères, polaires ou sphériques par exemple. Toutefois, dans ces deux situations la complexité des développements mathématiques augmente aussi considérablement. Une autre approche est alors envisageable : l'approche lagrangienne.

L'approche lagrangienne ne consiste pas à mettre explicitement en relation les déplacements et les forces mais à relier les énergies cinétiques et potentielles. L'équation de Lagrange est une reformulation du principe de Newton basée sur l'opérateur lagrangien, introduit au XVIII^e siècle par Joseph-Louis Lagrange [Penrose, 2007]. Un corps en mouvement dont la configuration peut être représentée par une variable q est alors régi dans un repère galiléen par l'égalité :

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{q}} \right) = \frac{\partial \mathcal{L}}{\partial q} \quad (1.2)$$

où la notation $\dot{\bullet}$ équivaut à $\frac{d\bullet}{dt}$ et où \mathcal{L} est le lagrangien du corps étudié, défini comme la différence entre son énergie cinétique et son énergie potentielle.

Alors que les modèles newtoniens (1.1) font apparaître directement les forces dans le modèle du robot, les modèles lagrangiens introduisent quant à eux les concepts d'énergies cinétiques et potentielles. Cette approche offre en pratique une plus grande liberté de représentation et peut simplifier le développement mathématique nécessaire à la modélisation du robot. Elle est notamment utilisée pour modéliser les robots à tubes concentriques [Renda et al., 2021, Pourafzal et al., 2021], les robots à câbles [Tummers et al., 2023] ainsi que les robots agonistes-antagonistes [Childs and Rucker, 2023, Tummers et al., 2024].

1.2.4 Utilisation des modèles

Les modèles sont utilisés pour de nombreuses applications en robotique continue et notamment pour développer des algorithmes de commande [Chikhaoui and Rosa, 2022]. Ces derniers sont divisés en deux types : lorsque l'état du robot est estimé exclusivement grâce au modèle, on parle de commande en *boucle ouverte*, et lorsqu'une information supplémentaire sur l'état du système est mesurée pour ajuster la commande du robot, on parle alors de commande en *boucle fermée*.

Les algorithmes les plus communs sont ceux conçus pour faire suivre une trajectoire à l'effecteur du robot, en boucle ouverte [Girerd and Morimoto, 2020] ou en boucle fermée [Alqumsan et al., 2019, Boehler et al., 2020]. Pour certaines applications, le contrôle de l'effecteur peut être soumis à des contraintes supplémentaires comme par exemple éviter des collisions entre deux robots utilisés simultanément [Sabetian et al., 2019], assurer un déploiement à la file indienne du robot [Gao et al., 2019, Girerd et al., 2020] ou encore éviter des situations mécaniquement instables [Khadem et al., 2019].

Les algorithmes de commande ne servent pas uniquement à contrôler la forme des robots continus : ils peuvent également être utilisés pour contrôler les forces de contact entre le robot et son environnement. Il est par exemple possible de minimiser la force de contact appliquée au bout du robot pour faciliter son déploiement dans le système cardiovasculaire [Fagogenis et al., 2019] ou encore de contrôler la force exercée lors d'une ablation ou d'une suture [Yasin and Simaan, 2021].

Ces algorithmes de commande qui permettent le contrôle des forces d'interaction nécessitent d'équiper les robots avec des capteurs de force, généralement placés au bout du robot [Yip et al., 2010] ou le long du robot [Arabagi et al., 2015]. Pour augmenter la miniaturisation des robots chirurgicaux, il est possible d'utiliser les modèles pour déporter ces capteurs dans une zone moins contrainte du robot [Du et al., 2024], utiliser d'autres types de capteurs, comme des capteurs de forme, moins encombrants [Qiao et al., 2021] ou même les positionner à l'extérieur du robot [Aloi et al., 2022, Ferguson et al., 2024].

1.3 Contributions et organisation du manuscrit

Au cours de la [section 1.2](#), nous avons vu que la modélisation des robots continus est un sujet d'étude vaste et actif, incontournable pour développer des algorithmes de commande plus sûrs et plus performants pour les robots chirurgicaux. Les travaux de cette thèse portent sur la modélisation des robots continus utilisés en chirurgie minimalement invasive. Nous verrons comment construire et utiliser des modèles cinéostatiques, analytiques et newtoniens de robots continus soumis à des interactions mécaniques. Les modèles présentés seront systématiquement développés pour les robots continus en général, puis dans le cas particulier des robots à tubes concentriques.

Le [chapitre 2](#) vise à clarifier les travaux de la littérature concernant la modélisation cinéostatique directe des robots continus. Ce chapitre, bien qu'il n'apporte pas de contribution en tant que telle, est toutefois essentiel pour définir les bases utiles dans les chapitres suivants.

Le [chapitre 3](#) présente la contribution centrale de ces travaux : un nouveau modèle pour définir la compliance des robots continus à travers *la matrice de compliance généralisée*. Cette nouvelle définition est exposée, puis une méthode permettant le calcul de la matrice de compliance généralisée est détaillée et validée numériquement.

Le [chapitre 4](#) démontre l'intérêt de cette nouvelle définition pour estimer les forces d'interaction des robots continus avec leur environnement. Nous formulons le problème d'estimation de forces par mesure de forme, puis une méthode de résolution est présentée, validée numériquement et testée expérimentalement sur un prototype de robot à tubes concentriques.

Le [chapitre 5](#) présente les conclusions des travaux effectués, puis dessine de nouvelles perspectives de recherche relatives à la modélisation de la compliance des robots continus ainsi qu'à ses potentielles utilisations pratiques.

1.4 Communications scientifiques

Les contributions présentées dans ce manuscrit ont fait l'objet de deux publications, la première dans une revue scientifique internationale, l'autre en présentation orale dans un symposium international :

- **Guillaume Lods**, Benoit Rosa, Bernard Bayle et Florent Nageotte, Exact derivative propagation method to compute the generalized compliance matrix for continuum robots : Application to concentric tubes continuum robots, *Mechanism and Machine Theory*, Volume 200, 15 September 2024, 105696.
DOI : <https://doi.org/10.1016/j.mechmachtheory.2024.105696>
Lien HAL : <https://hal.science/hal-04626541/>
- **Guillaume Lods**, Florent Nageotte, Bernard Bayle et Benoit Rosa, Force Sensing along Continuum Robots using Compliance Matrix and Shape Acquisition, *2024 Hamlyn Symposium on Medical Robotics* (présentation orale).
Lien Hamlyn Symposium : <https://www.hamlynsymposium.org/proceedings/>
Lien HAL : <https://hal.science/hal-04691006>

La première publication présente les contributions associées à la nouvelle définition de la matrice de compliance généralisée et à sa méthode de calcul. Le code Matlab utilisé pour calculer la matrice de compliance généralisée ainsi que sa version compilée sous forme de fichiers MEX sont partagés en ligne sous licence open source, afin d'en faire bénéficier la communauté de la robotique continue.

Lien GitHub : <https://github.com/benoitrosa/Generalized-Compliance-Computation-for-Continuum-Robots>

La seconde publication présente les contributions associées à l'utilisation de la matrice de compliance généralisée pour l'estimation des efforts d'interaction entre le robot continu et son environnement. Elle a donné lieu à une présentation orale à Londres le 27 juin 2024.

1.5 Financements

Ce travail a été financé par des fonds de l'Etat français gérés par l'Agence Nationale de la Recherche (ANR), subvention ANR-18-CE19-0012 (MACROS). Les recherches en robotique médicale du laboratoire ICube sont partiellement soutenues par les subventions ANR-10-IAHU-02 (IHU Strasbourg), ANR-11-LABX-0004 (Labex CAMI), ANR-10-IDEX-0002 (ITI HealthTech) et ANR-21-ESRE-0015 (TIRREX) dans le cadre du Programme d'Investissements d'Avenir.

Chapitre 2

Modélisation cinétostatique directe des robots continus

Nous avons vu précédemment que modéliser les robots continus est essentielle pour pouvoir les piloter correctement. Dans ce chapitre, nous allons présenter un modèle cinétostatique direct et newtonien permettant de prédire la forme de ces robots. Bien que la modélisation implique d'être adaptée à chaque type de robot, des généralités valables pour l'ensemble des robots continus seront décrites dans la [section 2.1](#) puis le cas spécifique des RTC sera traité dans la [section 2.2](#). Dans la [section 2.1](#), nous formulerons des hypothèses pour obtenir des équations permettant de décrire la forme et la mécanique des robots continus. Dans la [section 2.2](#), le modèle présenté sera appliqué au cas des RTC, ce qui nécessitera de formuler des hypothèses et équations spécifiques à ce type de robot. Enfin, nous présenterons quelques exemples de résultats obtenus grâce à cette modélisation. Ce chapitre ne présente pas de contribution à part entière mais se destine à ré-interpréter et clarifier des travaux pré-existants comme ceux de [[Lock et al., 2010](#)] et [[Rucker et al., 2010](#)] réalisés sur les RTC. Les contributions qui seront présentées dans le [chapitre 3](#) sont basées sur cette modélisation. C'est pourquoi une attention particulière sera portée à détailler les concepts, les hypothèses, et les équations du modèle cinétostatique direct.

Les opérateurs mathématiques, les symboles ainsi que les variables utilisées dans ce chapitre sont regroupés en annexe dans les tableaux [5.6](#) à [5.10](#).

2.1 Généralités sur les robots continus

Nous présentons dans cette section des notions et des équations génériques à tous les robots continus. La [sous-section 2.1.1](#) proposera une description différentielle de la forme, la [sous-section 2.1.2](#) un paramétrage des efforts appliqués sur le robot et de son actionnement puis la [sous-section 2.1.3](#) des hypothèses relatives aux types de déformations des robots continus. Nous verrons ensuite dans la [sous-section 2.1.4](#) comment utiliser les équations de Cosserat pour formuler le modèle cinétostatique comme un problème d'intégration. Finalement, une méthode de résolution de ce problème sera présentée dans la [sous-section 2.1.5](#).

2.1.1 Géométrie différentielle d'un robot continu

La description usuelle de la forme d'un robot série rigide consiste à décrire la position dans l'espace des points d'intérêt du robot, c'est-à-dire de la base, des articulations et de l'effecteur. Dans ce cas, décrire la forme du robot consiste à décrire un nombre fini de repères géométriques, généralement placés selon une convention comme celle de Denavit-Hartenberg [Denavit and Hartenberg, 1955]. Par la suite, si cela est nécessaire, il sera alors possible de décrire le reste du robot à partir des relations géométriques rigides reliant ces repères. Dans le cas d'un robot continu, il n'y a plus de tels points de référence hormis l'origine et l'effecteur du robot. La description géométrique doit évoluer pour passer d'un nombre fini de repères à une infinité de repères, répartis le long du robot. On introduit alors l'abscisse curviligne du robot, notée $s \in [0, L_0]$, où L_0 indique la longueur curviligne totale du robot. A chaque valeur de s , on associera un repère flottant orthonormé, noté $\mathcal{R}_0(s)$ et défini par (2.1), pour lequel le vecteur unitaire $\vec{z}_0(s)$ sera systématiquement orienté le long de la tangente du robot et dans le sens des s croissants :

$$\forall s \in [0, L_0], \mathcal{R}_0(s) = (s, \vec{x}_0(s), \vec{y}_0(s), \vec{z}_0(s)) \quad (2.1)$$

Cette représentation permet une description continue du robot tout en définissant les repères liés aux deux extrémités : le repère à l'origine $\mathcal{R}_0(s = 0)$, aussi noté \mathcal{R}_{B_0} , et le repère de l'effecteur $\mathcal{R}_0(s = L_0)$. En pratique, l'abscisse curviligne s pourra être discrétisée en des valeurs discrètes et on définira N_P comme le nombre total de points.

L'approche la plus naturelle pour décrire la forme du robot est d'utiliser la matrice de transformation rigide, notée $T_0(s) \in SE(3)$, qui représente la transformation entre \mathcal{R}_{B_0} et $\mathcal{R}_0(s)$. La matrice $T_0(s)$, définie par (2.2), est composée de la matrice de rotation $R_0(s) \in SO(3)$ et du vecteur de translation $p_0(s) \in \mathbb{R}^3$:

$$T_0(s) = \begin{bmatrix} R_0(s) & p_0(s) \\ 0_{1 \times 3} & 1 \end{bmatrix} \quad (2.2)$$

Cette matrice de transformation permet de décrire complètement la pose (i.e. la position et l'orientation) du repère flottant $\mathcal{R}_0(s)$ par rapport au repère de base \mathcal{R}_{B_0} . Pour décrire de manière continue la matrice $T_0(s)$, une description géométrique différentielle semble particulièrement adaptée. Pour cela on introduit la notation $\dot{\bullet} = \frac{d\bullet}{ds}$, largement utilisée dans le reste de ces travaux, pour indiquer la dérivation par rapport à s . On définit alors deux variations infinitésimales pour décrire $T_0(s)$: la variation d'angle (aussi appelée courbure), notée $u_0(s) \in \mathfrak{so}(3)$, et la variation de position, notée $v_0(s) \in \mathbb{R}^3$. Ces deux définitions sont énoncées par (2.3) et illustrées par la figure 2.1.

$$u_0(s) = \dot{R}_0(s)^\vee \quad (2.3a)$$

$$v_0(s) = \dot{p}_0(s) \quad (2.3b)$$

où \vee (aussi noté \bullet^\vee) désigne l'opérateur inverse de \wedge (aussi noté \bullet^\wedge) qui est l'opérateur asymétrique associé à $SO(3)$ et étendu à $SE(3)$ [Murray et al., 1994]. Ce dernier est défini par l'équation (2.4).

$$\forall u = \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} \in \mathbb{R}^3, \forall v = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} \in \mathbb{R}^3$$

$$\hat{u} = \begin{bmatrix} 0 & -u_z & u_y \\ u_z & 0 & -u_x \\ -u_y & u_x & 0 \end{bmatrix}, \quad \begin{bmatrix} v \\ u \end{bmatrix}^\wedge = \begin{bmatrix} 0 & -u_z & u_y & v_x \\ u_z & 0 & -u_x & v_y \\ -u_y & u_x & 0 & v_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.4)$$

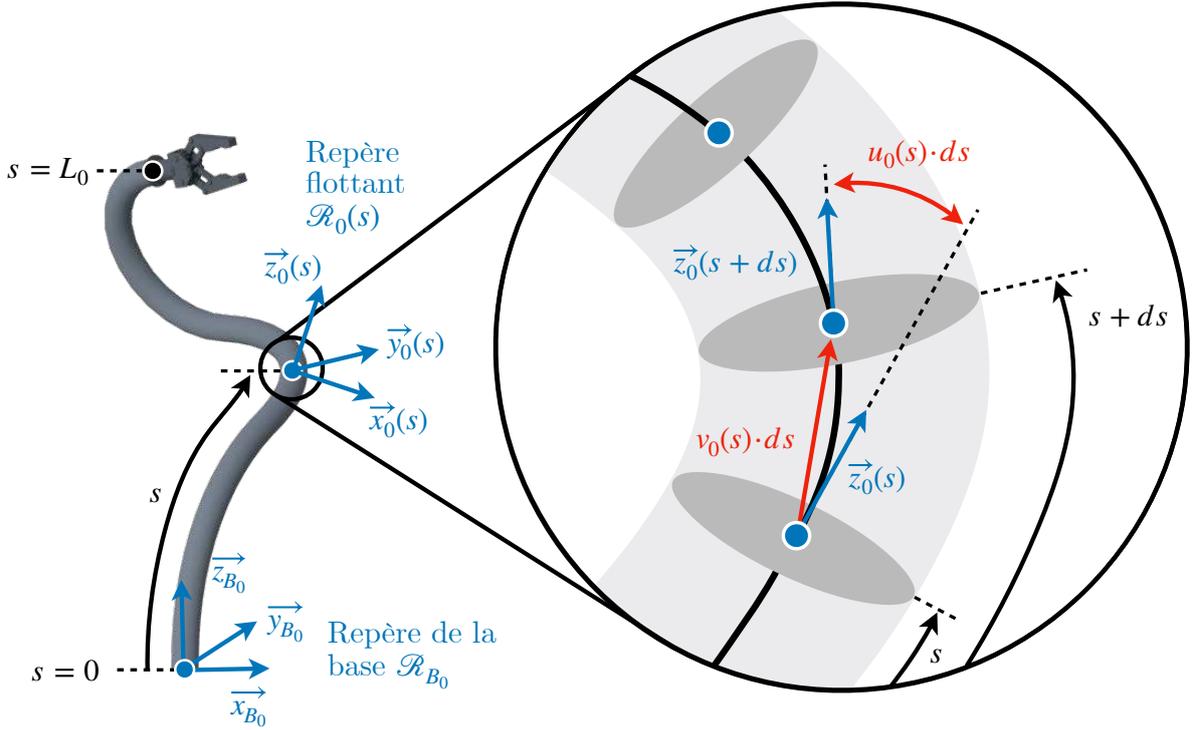


FIGURE 2.1 – Schéma illustrant les définitions de la variation de position $v_0(s)$ ainsi que de la courbure $u_0(s)$, utilisées pour décrire la forme d'un robot continu.

Les équations (2.3) ne sont valables telles quelles que lorsque les variables $u_0(s)$, $v_0(s)$, $R_0(s)$ et $p_0(s)$ sont toutes définies dans le même repère. En pratique, $u_0(s)$ et $v_0(s)$ seront définies dans le repère flottant $\mathcal{R}_0(s)$ tandis que $R_0(s)$ et $p_0(s)$ seront définies dans le repère de la base \mathcal{R}_{B_0} . En introduisant la notation $\bullet|_{\mathcal{R}}$ pour indiquer le repère \mathcal{R} de définition de la variable \bullet , il est alors possible de ré-écrire les équations (2.3) :

$$u_0(s)|_{\mathcal{R}_0(s)} = \left(R_0(s)^T \dot{R}_0(s) |_{\mathcal{R}_{B_0}} \right)^\vee \quad (2.5a)$$

$$v_0(s)|_{\mathcal{R}_0(s)} = R_0(s)^T \dot{p}_0(s) |_{\mathcal{R}_{B_0}} \quad (2.5b)$$

où \bullet^T est l'opérateur de transposition.

Afin de faciliter la lecture, on omettra les indications $\bullet|_{\mathcal{R}}$ dans les équations (2.5). Finalement, on obtient une description complète de la matrice de rotation $R_0(s)$ et du vecteur position $p_0(s)$ grâce aux deux descriptions différentielles :

$$\dot{R}_0(s) = R_0(s)\hat{u}_0(s) \quad (2.6a)$$

$$\dot{p}_0(s) = R_0(s)v_0(s) \quad (2.6b)$$

Une description équivalente mais plus compacte peut décrire la concaténation verticale de $v_0(s)$ et $u_0(s)$, notée $\xi_0(s) \in \mathfrak{se}(3)$:

$$\xi_0(s) = \begin{bmatrix} v_0(s) \\ u_0(s) \end{bmatrix} \quad (2.7)$$

Avec cette nouvelle description, les équations (2.5) et (2.6) peuvent se ré-écrire sous les formes respectives :

$$\xi_0(s) = \left(T_0(s)^{-1}T_0(s)\right)^\vee \quad (2.8a)$$

$$\dot{T}_0(s) = T_0(s)\hat{\xi}_0(s) \quad (2.8b)$$

2.1.2 Actionnement et interactions avec l'environnement du robot : sources de déformation

Le fonctionnement d'un robot continu consiste à contrôler sa forme en pilotant son actionnement. En fonction du type de robot considéré, l'actionnement peut être de nature très différente : l'actionnement des RC est la force de traction exercée sur les câbles, celui des RTC est la translation et la rotation des tubes et celui des RM est la modification d'un champ magnétique par exemple. L'actionnement comprend généralement plusieurs variables notées $q_i \in \mathbb{R}$ et regroupées en un unique vecteur noté $q \in \mathbb{R}^Q$.

L'actionnement n'est cependant pas la seule cause d'un changement de forme : les efforts générés lors d'un contact entre le robot et son environnement peuvent aussi être responsables de changements de forme. Ces interactions peuvent générer des forces, notées $f_0(s_0) \in \mathbb{R}^3$, et/ou des couples, notés $\tau_0(s_0) \in \mathbb{R}^3$, où $s_0 \in [0, L_0]$ indique l'abscisse curviligne du point d'application de ces efforts. On regroupera par la suite ces deux grandeurs en un unique vecteur noté $w_0(s_0)$ ¹ :

$$w_0(s_0) = \begin{bmatrix} \tau_0(s_0) \\ f_0(s_0) \end{bmatrix} \in \mathbb{R}^6 \quad (2.9)$$

Dans le cas d'un contact ponctuel, le point d'interaction est indiqué par son abscisse curviligne s_0 et lorsque le contact est réparti, la zone de contact est indiquée par l'intervalle $s_0 \in [l_{\min}, l_{\max}]$. Ces efforts, appliqués par l'environnement sur le robot, peuvent générer des efforts de réaction, internes au robot. On définit alors $n_0(s) \in \mathbb{R}^3$ et $m_0(s) \in \mathbb{R}^3$

1. La variable $w_0(s_0)$ fait référence au terme anglais "wrench" qui désigne la combinaison d'une force et d'un couple.

respectivement la force et le moment internes. Le lecteur notera que *les efforts internes ou externes* désignent à la fois *les forces et les moments (ou les couples) internes ou externes*. La [figure 2.2](#) regroupe les notations présentées, notamment les valeurs des abscisses curvilignes particulières, les repères ainsi que les efforts internes et externes.

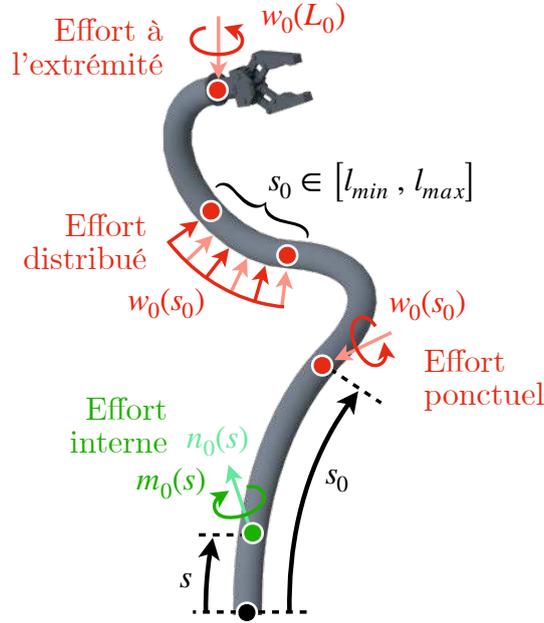


FIGURE 2.2 – Illustration d’un robot continu où sont représentés les efforts externes $w_0(s_0)$ ainsi que les efforts internes $m_0(s)$ et $n_0(s)$.

2.1.3 Modélisation des déformations du robot

Nous disposons désormais de variables pour décrire la forme du robot, son actionnement et les efforts d’interaction. Modéliser la forme du robot revient à proposer une relation mathématique associant les repères flottants $\mathcal{R}_0(s)$, l’actionnement q et les efforts externes $w_0(s_0)$. La modélisation proposée dans ce manuscrit est un assemblage de modèles mécaniques élémentaires issus d’hypothèses simplificatrices. La première hypothèse de modélisation consiste à modéliser la forme du robot continu par sa fibre neutre qui est une réduction abstraite de son volume en une ligne curviligne qui le parcourt dans toute sa longueur, supposée centrée par rapport à la section du robot. Il est ainsi admis que tous les modèles élémentaires utilisés peuvent être appliqués directement à la fibre neutre sans que le volume du robot n’ait de conséquences directes.

Bien que le robot continu soit une structure flexible, tous les types de déformations ne se manifestent pas de la même manière. En effet, les phénomènes de compression/élongation ou encore de cisaillement du robot sont très rares car ces derniers ont principalement lieu lorsque des efforts externes importants, relativement à la rigidité du robot, sont appliqués. Dans notre contexte d’utilisation de robots médicaux, il est admis que les efforts d’interaction ne seront pas si intenses : cela permet alors de simplifier le

modèle de déformation. Cette simplification revient à admettre l'hypothèse de Kirchhoff : on suppose que les déformations angulaires du robot sont bien plus importantes que la compression, l'élongation et le cisaillement, qui sont alors supposés négligeables. L'hypothèse de Kirchhoff, qui se traduit par l'égalité (2.10), est illustrée par la figure 2.3 sur laquelle on retrouvera les différents modes de déformation.

$$v_0(s) = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (2.10)$$

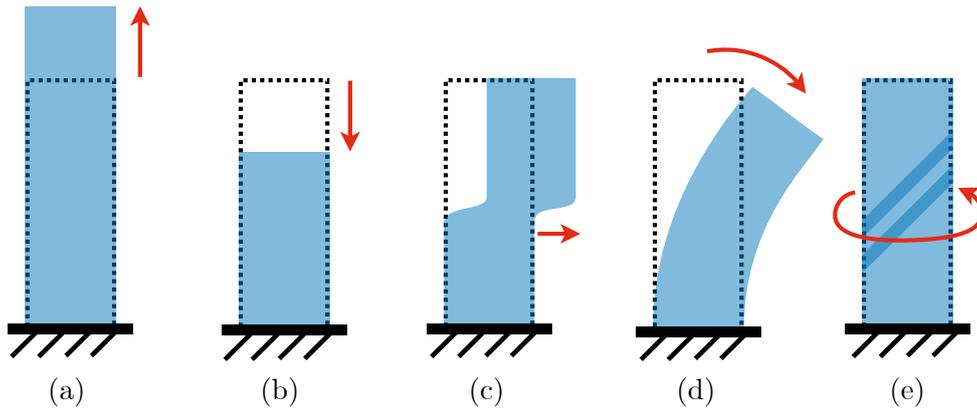


FIGURE 2.3 – Illustration des phénomènes d'élongation (a), de compression (b), de cisaillement (c) et des déformations angulaires de flexion (d) et de torsion (e).

Il apparaît alors nécessaire de définir un modèle pour prédire les déformations angulaires qui sont intrinsèquement dépendantes du matériau du robot. Chaque matériau admet des caractéristiques de déformations qui lui sont spécifiques et dont l'étude est une science à part entière. Les matériaux hyper-élastiques se distinguent par une plage de déformation élastique et linéaire très étendue. On admet alors que les déformations de notre robot resteront dans le domaine linéaire et élastique, qui peut toutefois dépasser les 10% de déformation dans le cas du nitinol [Stöckel, 1998]. On introduit alors la notation \bullet^* pour indiquer que la variable \bullet fait référence à un état initial, c'est-à-dire qui n'a pas subi de déformation. Avec cette notation, il devient alors possible de relier le changement de courbure d'une petite poutre flexible à un moment interne de flexion ou torsion mécanique grâce à l'équation (2.11) :

$$m_0(s) = K_0(s)(u_0(s) - u_0^*(s)) \quad (2.11)$$

où $K_0(s) \in \mathbb{R}^{3 \times 3}$ représente la matrice de raideur en flexion-torsion qui dépend des caractéristiques du matériau ainsi que de la géométrie du robot.

L'égalité (2.11) permet de représenter fidèlement la relation de cause-à-effet entre $m_0(s)$ et $u_0(s)$ dans le cas d'une petite poutre. Il convient de préciser qu'en raison de l'hypothèse de Kirchhoff, il n'est pas nécessaire de modéliser l'interaction entre $n_0(s)$ et $v_0(s)$ puisque que les phénomènes de compression/élongation et de cisaillement sont supposés négligeables.

2.1.4 Modélisation de la propagation des efforts internes le long du robot : les équations de Cosserat

Le modèle de déformation (2.11) est un élément clé de la modélisation de notre robot continu et flexible mais cette égalité ne peut s'appliquer que sur un élément de petit volume. Notre sujet d'étude, le robot continu, a quant à lui une forme semblable à une poutre, largement étudiée en Génie Civil et en étude des structures. Une poutre se caractérise par une dimension longitudinale bien plus grande que ses dimensions transversales. En raison de cette géométrie particulière, les efforts internes ne sont pas constants mais se propagent le long du robot. Nous proposons alors d'utiliser la théorie des frères Cosserat, développée spécialement pour modéliser la propagation d'efforts le long d'une poutre [Cosserat and Cosserat, 1909]. La théorie de Cosserat, initialement passée inaperçue, a été reprise dans un certain nombre d'études de mécanique théorique à partir des années 1950 [Cowin, 1962]. Elle est désormais largement utilisée pour formuler des modèles de robots continus, notamment grâce aux travaux de S.S. Antmann [Antman, 1995].

On considère ici un petit segment du robot, noté $\mathcal{S}(s)$, défini par l'intervalle $[s, s + ds]$ et délimité par les points $p_0(s)$ et $p_0(s + ds)$. Les efforts internes $m_0(s)$ et $n_0(s)$, observés sur le segment $\mathcal{S}(s)$, sont définis au point $p_0(s)$ et les efforts $m_0(s + ds)$ et $n_0(s + ds)$, observés sur le segment suivant $\mathcal{S}(s + ds)$, sont définis en $p_0(s + ds)$. Il est supposé également qu'un effort externe est systématiquement appliqué au bout du segment, c'est-à-dire qu'une force et un couple appliqués sur $\mathcal{S}(s)$ sont en réalité appliqués au point $p_0(s + ds)$. Les sections $\mathcal{S}(s)$ et $\mathcal{S}(s + ds)$ ainsi que les efforts internes et externes considérés sont illustrés sur la figure 2.4.

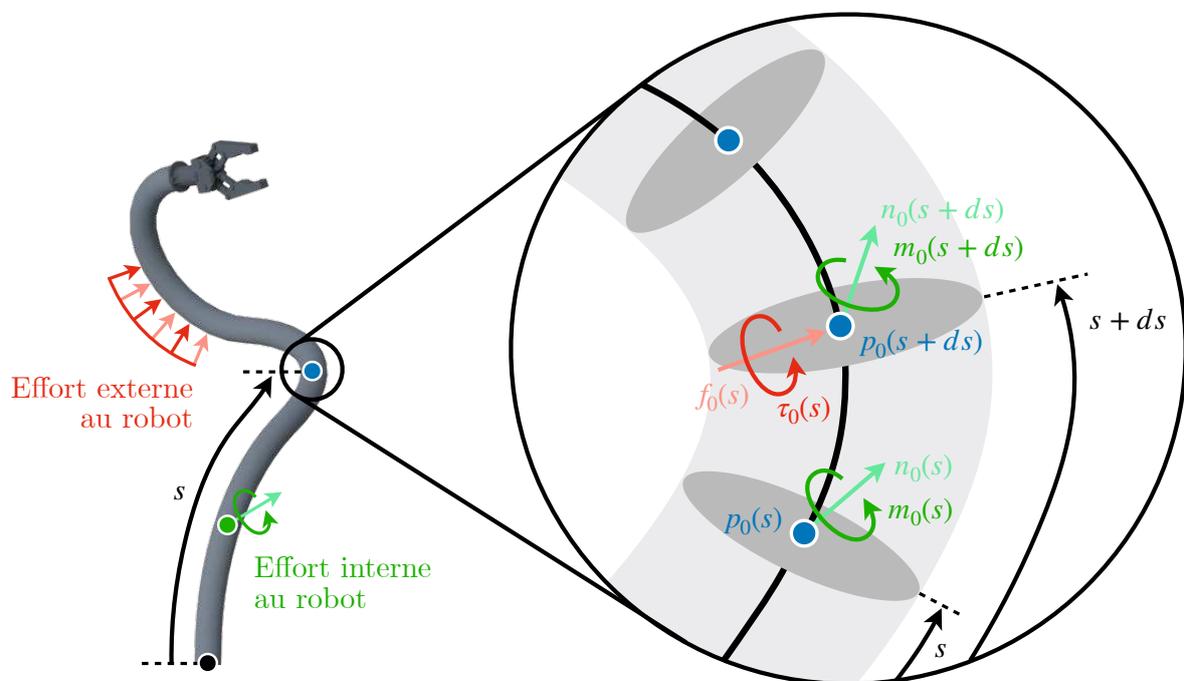


FIGURE 2.4 – Représentation des efforts appliqués à un segment infiniment petit.

En utilisant la première loi de Newton, il est alors possible de mettre en relation les différentes grandeurs présentées car l'équation (1.1) indique que si un corps est statique, ici notre section $\mathcal{S}(s)$, alors l'ensemble des forces et des moments appliqués sur ce dernier se compensent. Dans notre cas, cela revient à écrire les égalités (2.12) :

$$m_0(s) - m_0(s + ds) + \tau_0(s) + \begin{bmatrix} 0 \\ 0 \\ ds \end{bmatrix} \times \begin{pmatrix} f_0(s) - n_0(s + ds) \end{pmatrix} = 0_{3 \times 1} \quad (2.12a)$$

$$n_0(s) - n_0(s + ds) + f_0(s) = 0_{3 \times 1} \quad (2.12b)$$

où (2.12a) est l'égalité des moments exprimée au point $p_0(s)$ et où l'opérateur \times indique ici un produit vectoriel.

Il convient de préciser que les efforts externes $\tau_0(s+\epsilon)$ et $f_0(s+\epsilon)$ ne sont pas considérés dans les égalités (2.12) car ils sont appliqués sur le segment suivant $\mathcal{S}(s + ds)$ et non pas sur le segment étudié $\mathcal{S}(s)$. En substituant l'expression de $n_0(s + ds)$ donnée par (2.12b) dans l'équation (2.12a), il est possible de faire apparaître des taux d'accroissement ainsi que la variable $v_0(s)$ en divisant les deux membres des équations par ds :

$$\frac{m_0(s + ds) - m_0(s)}{ds} = -\frac{0_{3 \times 1} - \tau_0(s)}{ds} - \hat{v}_0(s)n_0(s) \quad (2.13a)$$

$$\frac{n_0(s + ds) - n_0(s)}{ds} = -\frac{0_{3 \times 1} - f_0(s)}{ds} \quad (2.13b)$$

Ces égalités ont pour vocation à être appliquées à des segments infiniment petits et donc à une représentation continue du robot. Ainsi, en appliquant la limite $ds \rightarrow 0$ sur ces équations, on obtient les relations différentielles suivantes :

$$\dot{m}_0(s) = -\dot{\tau}_0(s) - \hat{v}_0(s)n_0(s) \quad (2.14a)$$

$$\dot{n}_0(s) = -\dot{f}_0(s) \quad (2.14b)$$

Les équations (2.14) permettent de modéliser la propagation des efforts internes $m_0(s)$ et $n_0(s)$ le long du robot et en fonction des efforts externes $\tau_0(s)$ et $f_0(s)$. Ces égalités sont implicitement définies dans un repère unique pour tout le robot, or les courbures $u_0(s)$ et $v_0(s)$ sont quant à elles définies localement dans le repère flottant $\mathcal{R}_0(s)$. Il faut alors les compléter pour inclure les variations du repère $\mathcal{R}_0(s)$. En rappelant que $\bullet|_{\mathcal{R}}$ précise que la variable \bullet est exprimée dans le repère \mathcal{R} et en notant $\binom{s+ds}{s} R_s$ la matrice de rotation pour passer de l'orientation de $\mathcal{R}_0(s)$ à celle de $\mathcal{R}_0(s + ds)$, on obtient les changements de repères (2.15) :

$$m_0(s)|_{\mathcal{R}_0(s+ds)} = \binom{s+ds}{s} R_s m_0(s)|_{\mathcal{R}_0(s)} \quad (2.15a)$$

$$n_0(s)|_{\mathcal{R}_0(s+ds)} = \binom{s+ds}{s} R_s n_0(s)|_{\mathcal{R}_0(s)} \quad (2.15b)$$

Ces égalités sont ensuite dérivées par rapport à s et par rapport au repère local $\mathcal{R}_0(s)$. En utilisant l'équation (2.6a) appliquée au repère $\mathcal{R}_0(s + ds)$, on obtient alors :

$$\dot{m}_0(s)\Big|_{\mathcal{R}_0(s+ds)} = \left({}^{s+ds}R_s\right) \hat{u}_0(s)\Big|_{\mathcal{R}_0(s+ds)} m_0(s)\Big|_{\mathcal{R}_0(s)} + \left({}^{s+ds}R_s\right) \dot{m}_0(s)\Big|_{\mathcal{R}_0(s)} \quad (2.16a)$$

$$\dot{n}_0(s)\Big|_{\mathcal{R}_0(s+ds)} = \left({}^{s+ds}R_s\right) \hat{u}_0(s)\Big|_{\mathcal{R}_0(s+ds)} n_0(s)\Big|_{\mathcal{R}_0(s)} + \left({}^{s+ds}R_s\right) \dot{n}_0(s)\Big|_{\mathcal{R}_0(s)} \quad (2.16b)$$

En appliquant à nouveau la limite $ds \rightarrow 0$, on obtient $\left({}^{s+ds}R_s\right) \xrightarrow{ds \rightarrow 0} I_{3 \times 3}$. De plus, en substituant respectivement les termes $\dot{m}_0(s)\Big|_{\mathcal{R}_0(s)}$ et $\dot{n}_0(s)\Big|_{\mathcal{R}_0(s)}$ par (2.14a) et (2.14b), on obtient alors les expressions de propagations suivantes :

$$\dot{m}_0(s)\Big|_{\mathcal{R}_0(s)} = \hat{u}_0(s)\Big|_{\mathcal{R}_0(s)} m_0(s)\Big|_{\mathcal{R}_0(s)} - \dot{\tau}_0(s) - \hat{v}_0(s) n_0(s) \quad (2.17a)$$

$$\dot{n}_0(s)\Big|_{\mathcal{R}_0(s)} = \hat{u}_0(s)\Big|_{\mathcal{R}_0(s)} n_0(s)\Big|_{\mathcal{R}_0(s)} - \dot{f}_0(s) \quad (2.17b)$$

Ces nouvelles équations (2.17) prennent désormais en compte l'évolution du repère flottant $\mathcal{R}_0(s)$ tandis que les équations initiales (2.14) n'étaient valables que pour un repère fixe. Désormais toutes les variables, à l'exception de $\dot{\tau}_0(s)$ et $\dot{f}_0(s)$, sont référencées par rapport au repère flottant $\mathcal{R}_0(s)$, c'est pourquoi les notations $\bullet\Big|_{\mathcal{R}_0(s)}$ seront omises pour simplifier la lecture. En pratique, il est plus simple d'exprimer les efforts externes $\tau_0(s)$ et $f_0(s)$ par rapport au repère de la base \mathcal{R}_{B_0} plutôt que par rapport au repère flottant $\mathcal{R}_0(s)$. Pour cette raison, la matrice de rotation $R_0(s)$ est ajoutée devant les termes $\dot{\tau}_0(s)$ et $\dot{f}_0(s)$, exprimés systématiquement dans le repère \mathcal{R}_{B_0} . Finalement, le développement présenté abouti aux équations de Cosserat appliquées aux robots continus (2.18) qui permettent de relier les déformations du robot avec ses efforts internes et externes :

$$\dot{m}_0(s) = \hat{u}_0(s) m_0(s) - \hat{v}_0(s) n_0(s) - R_0(s) \dot{\tau}_0(s) \quad (2.18a)$$

$$\dot{n}_0(s) = \hat{u}_0(s) n_0(s) - R_0(s) \dot{f}_0(s) \quad (2.18b)$$

D'autres équations modélisant le comportement des poutres existent comme celles de Euler-Bernoulli, largement utilisées en Génie Civil. Contrairement aux équations de Euler-Bernoulli, les équations de Cosserat ne nécessitent pas l'hypothèse de "petites" déformations. Pour cette raison, les équations (2.18) obtenues grâce à la théorie de Cosserat peuvent être considérées comme géométriquement exactes.

2.1.5 Construction d'un vecteur d'état et résolution du modèle différentiel

On dispose désormais de plusieurs égalités pouvant décrire les déformations du robot. Les équations (2.18) relient les efforts internes du robot aux efforts externes ainsi qu'à la courbure tandis que les équations (2.6) relient la courbure à la forme du robot. Il semble alors nécessaire d'introduire l'actionnement q dans la modélisation. Puisque le type d'actionnement dépend intrinsèquement du type de robot continu étudié, il n'est pas

envisageable de proposer un modèle général incluant l'actionnement q et valable pour tous les robots continus. Un cas d'étude, les RTC, sera toutefois détaillé dans la [section 2.2](#) et pour lequel l'actionnement sera relié aux courbures.

Toutes les variables décrites à partir d'une relation différentielle peuvent ensuite être regroupées dans un vecteur d'état, noté $y(s) \in \mathbb{R}^Y$, contenant à minima les variables $R_0(s)$ et $p_0(s)$. On retrouve dans la littérature des formulations du vecteur d'état contenant les efforts internes $m_0(s)$ et $n_0(s)$ [[Lock et al., 2010](#)] ou d'autres contenant les grandeurs $u_0(s)$ et $v_0(s)$ à la place [[Rucker et al., 2010](#)]. L'introduction d'un vecteur d'état permet de regrouper les différentes équations différentielles dans une unique relation différentielle de la forme :

$$\dot{y}(s) = \mathcal{F}(y, s) \quad (2.19)$$

Puisque l'équation (2.19) contient à minima les équations (2.6), résoudre le modèle cinéto-statique du robot revient donc à intégrer l'équation globale (2.19). Comme pour tout problème d'intégration, il est nécessaire de disposer d'un minimum d'information aux limites pour pouvoir procéder à l'intégration. On distingue alors deux situations différentes :

- L'intégralité des valeurs du vecteur d'état $y(s)$ est connue à l'une des extrémités du robot $s \in \{0, L_0\}$. Dans ce cas, l'équation (2.19) peut être intégrée en une fois et on parle alors d'un Problème d'intégration aux Valeurs Initiales (PVI²). C'est le cas des poutres de Cosserat, encastées à leur base et non soumises à un effort extérieur.
- Les valeurs connues des conditions aux limites sont réparties entre plusieurs points de l'intervalle d'intégration. Dans cette situation, indépendamment du sens d'intégration du problème (de $s = 0 \rightarrow L_0$ ou inversement), au moins l'une des valeurs initiales de $y(s)$ ne sera pas connue : on parle alors d'un Problème d'intégration aux Valeurs Limites (PVL³). C'est notamment le cas pour les modèles des RTC, des RAA ou encore des RC. Pour le cas des RC par exemple, la pose $T_0^y(0)$ à la base du robot est connue par définition du repère \mathcal{R}_{B_0} . Hors cas particulier [[Bentley et al., 2022](#)], il n'est cependant pas possible de déterminer directement ni la courbure $u_0(0)$, ni la variation de position $v_0(0)$, ni les efforts internes $m_0(0)$ et $n_0(0)$ à l'extrémité proximale, c'est-à-dire à la base du robot $s = 0$. Il faut donc dans ce cas utiliser l'hypothèse d'équilibre statique à l'extrémité distale, c'est-à-dire au bout du robot $s = L_0$, qui implique notamment les efforts internes $m_0(L_0)$ et $n_0(L_0)$.

Sachant que les PVL sont plus complexes à résoudre que les PVI, il est envisageable d'instrumenter la base du robot avec un capteur de forces pour mesurer directement les efforts internes et simplifier le PVL ou même, dans le cas d'une poutre de Cosserat (non actionnée), se ramener directement à un PVI [[Xu and Patel, 2012](#)]. Les PVL ont été largement étudiés dans la littérature et la méthode de résolution la plus couramment

2. "Initial Value Problem" (IVP) en anglais.

3. "Boundary Value Problem" (BVP) en anglais.

utilisée est la Méthode par Tirages (MT)⁴. En sélectionnant un sens d'intégration (ici de $s = 0 \rightarrow L_0$), il est possible de regrouper les coefficients du vecteur d'état $y(s)$ qui ne sont pas connus à la base $s = 0$ en un sous-vecteur d'état noté $y_u(s) \in \mathbb{R}^U$. On notera également $y_k(s) \in \mathbb{R}^{Y-U}$ le sous-vecteur d'état complémentaire⁵ :

$$y(s) = \begin{bmatrix} y_u(s) \\ y_k(s) \end{bmatrix} \quad (2.20)$$

Puisque la valeur de $y_u(0)$ est inconnue au départ de l'intégration, la MT lui attribue une valeur arbitraire. En réalisant l'intégration de l'équation (2.19), nous obtenons alors deux valeurs différentes de $y_u(L_0)$: l'une résultant de l'intégration et une autre issue des conditions aux limites connues non utilisées. La comparaison de ces deux valeurs peut être représentée par un vecteur (aussi appelé résidu) noté $b \in \mathbb{R}^B$. La MT procède alors à une optimisation itérative sur $y_u(0)$ à partir de la valeur de b . Pour une MT utilisant un algorithme de Levenberg-Marquardt [Levenberg, 1944], cette optimisation nécessitera de connaître la jacobienne du problème d'optimisation $\frac{\partial b}{\partial y_u(0)}$. Cette quantité permettra de faire converger b itérativement vers $0_{B \times 1}$ et par conséquent de retrouver les valeurs manquantes $y_u(0)$.

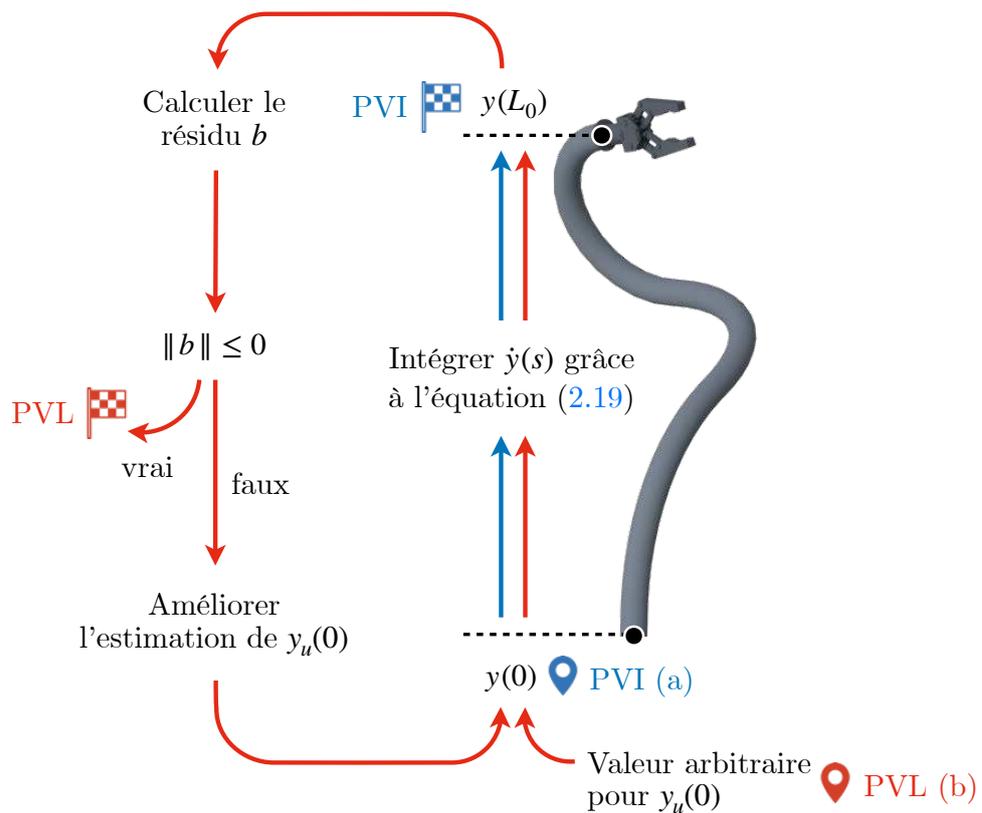


FIGURE 2.5 – Illustration de la résolution d'un PVI (a) et d'un PVL (b).

4. "Shooting Method" en anglais.

5. Les indices \bullet_u et \bullet_k font respectivement référence aux termes anglais "unknown" et "known".

Résoudre le PVL revient donc à intégrer le problème différentiel (2.19) tout en garantissant $b \approx 0_{B \times 1}$. La figure 2.5 illustre les méthodes de résolution des PVI et PVL. Une fois le PVL résolu, nous disposons alors d'une description complète de la forme du robot $p_0(s)$ qui dépend de plusieurs paramètres, organisés en deux groupes de variables :

- Ω_{rob} regroupe l'ensemble des caractéristiques constantes pour un même robot. Dans le cas des RTC, cela concerne notamment le nombre de tubes, la géométrie des tubes et les caractéristiques du matériau.
- Ω_{ctrl} regroupe l'ensemble des paramètres variables pour un même robot. Cela englobe les efforts externes $w_0(s_0)$ appliqués sur le robot ainsi que les variables d'actionnement q :

$$\Omega_{ctrl} = \{w_0(s_0), q\} \quad (2.21)$$

On définit alors la fonction $\mathcal{M}(\bullet)$ qui calcule la forme du robot $p_0(s)$ à partir des valeurs de Ω_{rob} et Ω_{ctrl} :

$$p_0(s) = \mathcal{M}(s, \Omega_{rob}, \Omega_{ctrl}) \quad (2.22)$$

Cette représentation haut-niveau de la résolution du modèle sera utile notamment dans le chapitre 3.

2.2 Cas d'étude : Modélisation cinétostatique d'un robot à tubes concentriques

Maintenant que les notions génériques à tous les robots continus ont été présentées, nous allons les adapter et les compléter pour traiter en détails le cas des RTC. Comme présenté dans la [section 1.1](#), les RTC sont composés de plusieurs tubes interagissant entre eux et permettant ainsi de modifier la forme globale du robot en pilotant la rotation et le déploiement des tubes. Dans cette partie, nous allons expliciter les équations permettant de développer un modèle de RTC à partir de la théorie de Cosserat ainsi que la méthode de résolution par tirages. Nous verrons dans la [sous-section 2.2.1](#) comment adapter le paramétrage de l'abscisse curviligne, des courbures et des repères puis dans la [sous-section 2.2.2](#) comment définir l'actionnement des RTC. Ensuite les hypothèses complémentaires et équations associées utiles pour décrire les interactions entre les tubes seront présentées dans la [sous-section 2.2.3](#) et la construction de l'équation différentielle sur le vecteur d'état sera détaillée dans la [sous-section 2.2.4](#). Enfin, nous verrons dans la [sous-section 2.2.5](#) comment résoudre le modèle cinétostatique direct et quelques résultats seront présentés dans la [sous-section 2.2.6](#).

2.2.1 Abscisse curviligne et translation des tubes

Dans le but de développer un modèle généralisable, on considère ici un RTC composé de $N_T \in \mathbb{N}^*$ tubes. Un indice $i \in \llbracket 1, N_T \rrbracket$ est attribué à chaque tube : l'indice $i = 1$ est attribué au tube le plus court puis croît jusqu'à $i = N_T$ qui est attribué au tube le plus long. L'indexage $i = 0$ est spécifiquement attribué au robot étendu (c'est-à-dire à l'assemblage des tubes). Les indices i seront utilisés à travers la notation \bullet_i qui permet de préciser que la variable \bullet est attachée au tube $i \in \llbracket 1, N_T \rrbracket$ ou au robot dans le cas \bullet_0 . Puisque tous les tubes n'ont pas forcément la même longueur déployée, nous introduisons la variable $N_T(s)$ pour représenter tous les indices i des tubes présents à une abscisse curviligne donnée s .

Les variables d'actionnement des tubes seront notées θ_{ci} et β_{ci} , désignant respectivement la rotation et le déploiement du tube i . Comme présenté dans la [sous-section 2.1.2](#), on rappelle que les variables articulaires θ_{ci} et β_{ci} peuvent être assemblées dans un unique vecteur d'actionnement $q \in \mathbb{R}^{2N_T}$:

$$q = \begin{bmatrix} \theta_c \\ \beta_c \end{bmatrix} = \begin{bmatrix} \theta_{c1} \\ \vdots \\ \theta_{cN_T} \\ \beta_{c1} \\ \vdots \\ \beta_{cN_T} \end{bmatrix} \quad (2.23)$$

Puisqu'une partie des tubes sera systématiquement non-déployée, l'abscisse curviligne s sera étendue sur un domaine négatif. On voit alors apparaître deux domaines distincts pour s : la partie actionnée ($s \leq 0$) et la partie déployée ($s \geq 0$). Les points particuliers de l'abscisse curviligne, qui dépendent des longueurs des tubes i , notées L_i , sont illustrés par la [figure 2.6](#) et listés ci-dessous :

- $s = 0_0 = 0_{N_T} \leq 0$ désigne l'extrémité proximale du robot (qui coïncide avec l'extrémité proximale du tube $i = N_T$)
- $s = 0_i = \beta_{ci} - L_i \leq 0$ désigne l'extrémité inférieure du tube i
- $s = 0$ indique le premier point de déploiement du robot qui est également l'origine du repère fixe de la base \mathcal{R}_{B_0}
- $s = \beta_{ci} \geq 0$ désigne l'extrémité supérieure du tube i
- $s = L_0 = \beta_{cN_T} \geq 0$ désigne l'extrémité distale du robot (qui coïncide avec l'extrémité distale du tube $i = N_T$)

Le lecteur remarquera que les extrémités des tubes sont paramétrées par la variable d'actionnement en translation β_{ci} . Pour assurer le bon fonctionnement du mécanisme d'actionnement, un tube ne peut pas être déployé d'une longueur supérieure à sa longueur ni d'une distance négative. Pour cette raison, on obtient la contrainte d'actionnement (2.24) :

$$\forall i \in \llbracket 1, N_T \rrbracket, 0 \leq \beta_{ci} \leq L_i \quad (2.24)$$

Plus tard dans la modélisation [sous-section 2.2.3](#), il sera supposé que les tubes ne présentent pas de jeu mécanique entre eux. Afin de s'approcher au mieux de cette hypothèse, la commande en translation des tubes assurera systématiquement que le tube $i \in [2, N_T - 1]$ soit en contact avec ses voisins $i - 1$ et $i + 1$. Cette contrainte de commande, associée à l'inégalité (2.24), permet de définir l'inégalité globale régissant l'actionnement en translation des tubes :

$$\beta_{cN_T} - L_{N_T} \leq \dots \leq \beta_{c1} - L_1 \leq 0 \leq \beta_{c1} \leq \dots \leq \beta_{N_T} \quad (2.25)$$

Maintenant que la définition de l'abscisse curviligne a été adaptée au cas des RTC et que la condition sur le déploiement (2.25) a été explicitée, il est alors possible d'étendre la définition (2.1) du repère $\mathcal{R}_0(s)$ sur l'intervalle négatif $s \in [0_0, 0]$. En plus du repère $\mathcal{R}_0(s)$ associé au robot, chaque tube i dispose également de son propre repère flottant, noté $\mathcal{R}_i(s)$. De manière analogue au repère \mathcal{R}_{B_0} , le repère $\mathcal{R}_{B_i} = \mathcal{R}_i(s = 0_i)$ est le repère attaché à la base du tube i .

$$\forall i \in \llbracket 1, N_T \rrbracket, \forall s \in [0_i, \beta_{ci}], \mathcal{R}_i(s) = \left(s, \vec{x}_i(s), \vec{y}_i(s), \vec{z}_i(s) \right) \quad (2.26)$$

Dans la suite de la modélisation, les repères $\mathcal{R}_i(s)$ seront notamment utilisés pour décrire la pré-courbure des tubes ainsi que les déformations en torsion.

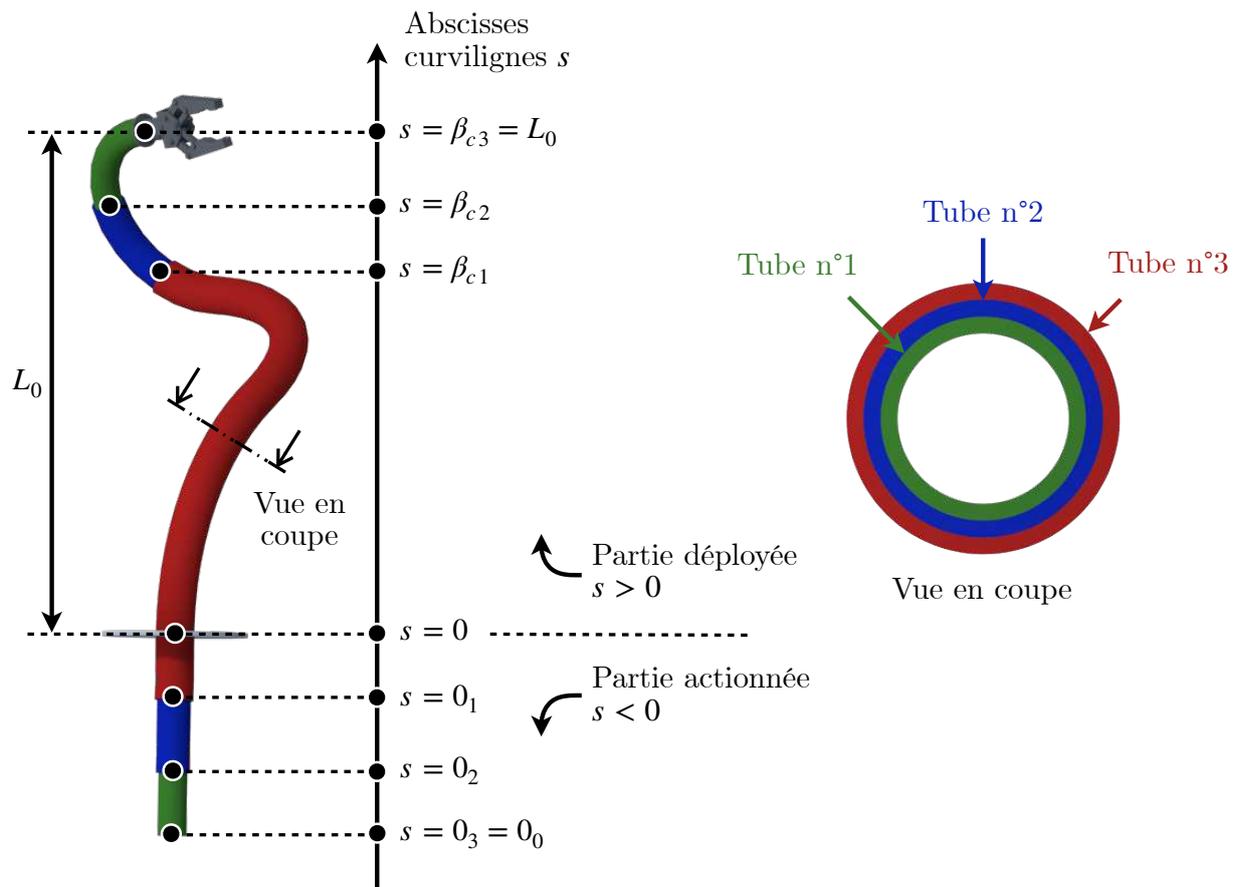


FIGURE 2.6 – Paramétrage des abscisses curvilignes particulières pour un RTC composé de 3 tubes.

2.2.2 Courbures et rotation des tubes

La pré-courbure des tubes du RTC joue un rôle essentiel dans les performances de commande comme l'espace de travail ou encore la stabilité mécanique du robot. Dans cette modélisation, il sera tout d'abord admis que les sections des tubes sont circulaires, constantes et centrées sur la fibre neutre sur toute la longueur des tubes⁶. Pour la description géométrique des tubes, illustrée par la [figure 2.7](#), on notera R_i et r_i les rayons respectivement extérieurs et intérieurs. Nous supposons également que les tubes i partagent tous le même type de forme, paramétrée suivant deux longueurs : une première rectiligne de longueur L_{ri} puis une seconde courbée de longueur L_{ci} . La partie courbée est supposée suivre une forme circulaire, de rayon de courbure R_{ci} , dans le plan $(\vec{y}_{B_i}, \vec{z}_{B_i})$. On notera que la longueur totale du tube i se trouve donc être la somme des longueurs rectilignes et courbées (2.27) :

$$L_i = L_{ri} + L_{ci} \quad (2.27)$$

6. Pour d'autres types de robots continus comme les RAA cette hypothèse n'est habituellement pas admise.

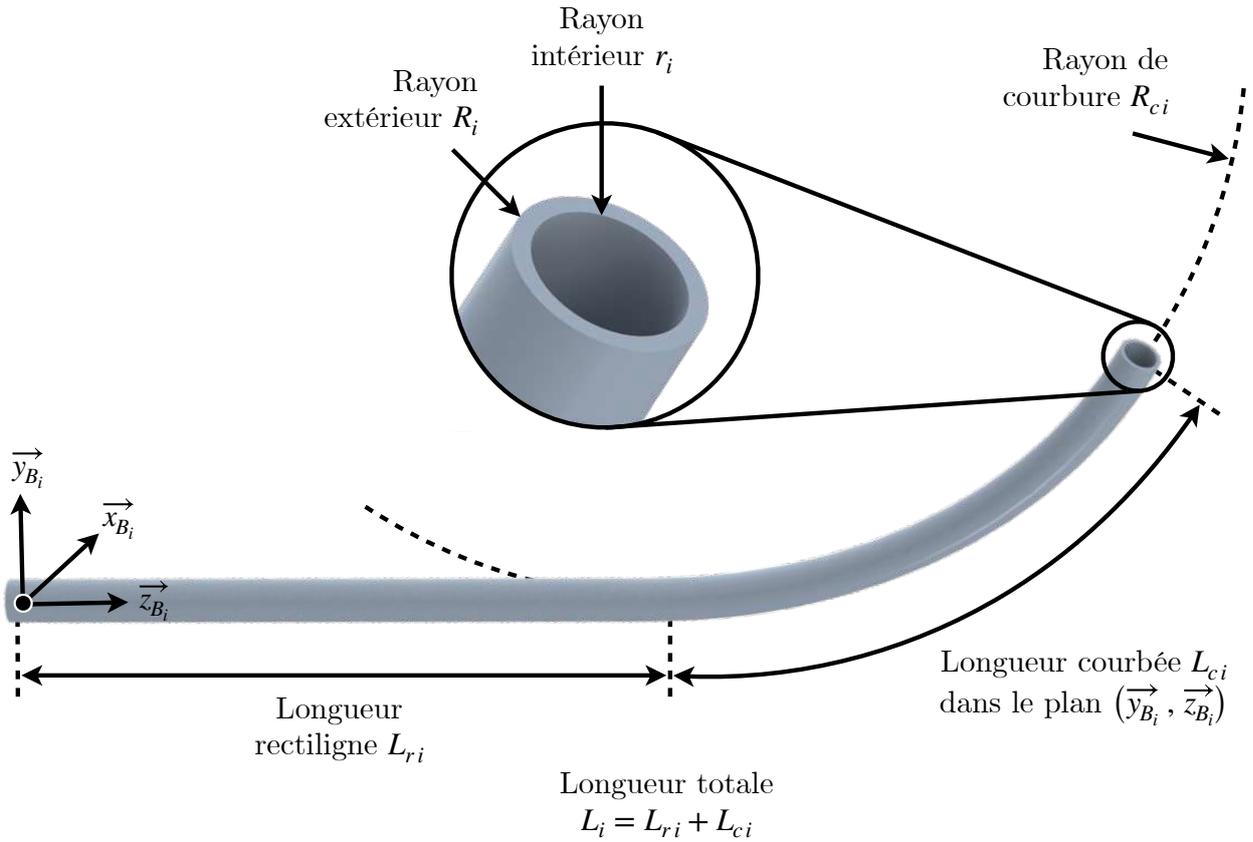


FIGURE 2.7 – Représentation du paramétrage géométrique des tubes.

Le paramétrage géométrique des tubes permet alors de décrire leur courbure. Or, puisque les tubes du RTC sont destinés à être déformés, on utilisera la notation \bullet^* pour distinguer la courbure initiale des tubes, notée $u_i^*(s) \in \mathbb{R}^3$, de la courbure déformée, notée $u_i(s) \in \mathbb{R}^3$. Ainsi, la courbure initiale $u_i^*(s)$ peut être définie explicitement par (2.28) et nous verrons par la suite comment estimer la courbure après déformation $u_i(s)$.

$$u_i^*(s) \Big|_{\mathcal{R}_{B_i}} = \begin{bmatrix} \phi_i(s) \\ 0 \\ 0 \end{bmatrix} \quad (2.28)$$

où

$$\phi_i(s) = \begin{cases} \frac{1}{R_{ci}} & \text{si } \max(0, (\beta_{ci} - L_{ci})) \leq s \leq \beta_{ci} \\ 0 & \text{sinon} \end{cases} \quad (2.29)$$

On remarquera que l'expression (2.28) fait intervenir la variable d'actionnement en translation β_{ci} et que le vecteur $u_i^*(s)$ est décrit dans le repère du tube \mathcal{R}_{B_i} et non dans le repère du robot \mathcal{R}_{B_0} . De plus, l'équation (2.29) implique que :

$$\forall s \leq 0, \phi_i(s) = 0 \quad (2.30)$$

2.2. Cas d'étude : Modélisation cinéto-statique d'un robot à tubes concentriques

L'égalité (2.30) est liée à une hypothèse de modélisation implicite qui suppose que les tubes restent parfaitement rectilignes dans la partie actionnée du RTC. En pratique, les guidages mécaniques intégrés au mécanisme d'actionnement des tubes permettent de se rapprocher au mieux de cette hypothèse, et cela même si une longueur initialement pré-courbée est présente dans le mécanisme. Pour pouvoir exprimer la courbure des tubes $u_i^*(s)$ dans le repère du robot, il est alors nécessaire de décrire la pose de \mathcal{R}_{B_i} par rapport à \mathcal{R}_{B_0} . C'est à ce moment qu'intervient la seconde variable d'actionnement : la rotation des tubes θ_{ci} définie par (2.31) :

$$\theta_{ci} = \alpha(\overrightarrow{x_{B_0}}, \overrightarrow{x_{B_i}}) = \alpha(\overrightarrow{y_{B_0}}, \overrightarrow{y_{B_i}}) \quad (2.31)$$

où l'opérateur $\alpha(\bullet_1, \bullet_2)$ représente l'angle signé entre les vecteurs \bullet_1 et \bullet_2 .

A partir de la définition (2.31), la courbure $u_i^*(s)$ peut alors être exprimée dans le repère \mathcal{R}_{B_0} en ajoutant une rotation de θ_{ci} autour de $\overrightarrow{z_{B_0}}$:

$$u_i^*(s) \Big|_{\mathcal{R}_{B_0}} = Rot_z(\theta_{ci}) \begin{bmatrix} \phi_i(s) \\ 0 \\ 0 \end{bmatrix} \quad (2.32)$$

où $Rot_z(\bullet) \in SO(3)$ représente la matrice de rotation d'un angle \bullet autour de $\overrightarrow{z_{B_0}}$:

$$Rot_z(\bullet) = \begin{pmatrix} \cos(\bullet) & -\sin(\bullet) & 0 \\ \sin(\bullet) & \cos(\bullet) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (2.33)$$

On remarque que la description de la courbure présentée par (2.32) est paramétrée par les deux variables d'actionnement θ_{ci} et β_{ci} . Sachant que le vecteur $u_i^*(s)$ sera exploité pour la suite de la modélisation, on comprend alors que la forme du robot sera indirectement paramétrée par l'actionnement q (2.23) à travers $u_i^*(s)$.

Il a été présenté dans la sous-section 2.1.3 que la modélisation proposée prend en compte les phénomènes de torsion. Puisque les repères flottants $\mathcal{R}_i(s)$ sont attachés aux tubes, il semble naturel d'utiliser le repère $\mathcal{R}_0(s)$ comme référence pour quantifier la torsion. Dans cet objectif, l'orientation de repère $\mathcal{R}_0(s)$ est alors contrainte selon les caractéristiques d'un repère de Bishop : le repère $\mathcal{R}_0(s)$ garantit une rotation nulle autour de son axe $\overrightarrow{z_0}(s)$. Cette orientation particulière se traduit par la contrainte (2.34) sur la courbure $u_0(s)$:

$$\forall s \in [0_0, L_0], u_0|_z(s) = 0 \quad (2.34)$$

où la notation $\bullet|_{x/y/z}$ spécifie une restriction du vecteur \bullet à l'une ou plusieurs de ses composantes, portées par les vecteurs unitaires $\overrightarrow{x_0}(s)$, $\overrightarrow{y_0}(s)$ ou $\overrightarrow{z_0}(s)$.

Maintenant que l'orientation du repère $\mathcal{R}_0(s)$ par rapport à son vecteur $\overrightarrow{z_0}(s)$ est fixe, il est alors possible de définir l'angle de torsion du tube i , noté $\theta_i(s)$, par :

$$\forall s \in [0_i, \beta_{ci}], \theta_i(s) = \alpha(\overrightarrow{x_0}(s), \overrightarrow{x_i}(s)) - \theta_{ci} \quad (2.35)$$

Les repères flottants $\mathcal{R}_0(s)$ et $\mathcal{R}_i(s)$, les angles de torsion $\theta_i(s)$ et les angles d'actionnement θ_{ci} sont illustrés par la figure 2.8, pour l'exemple d'un RTC composé de 2 tubes.

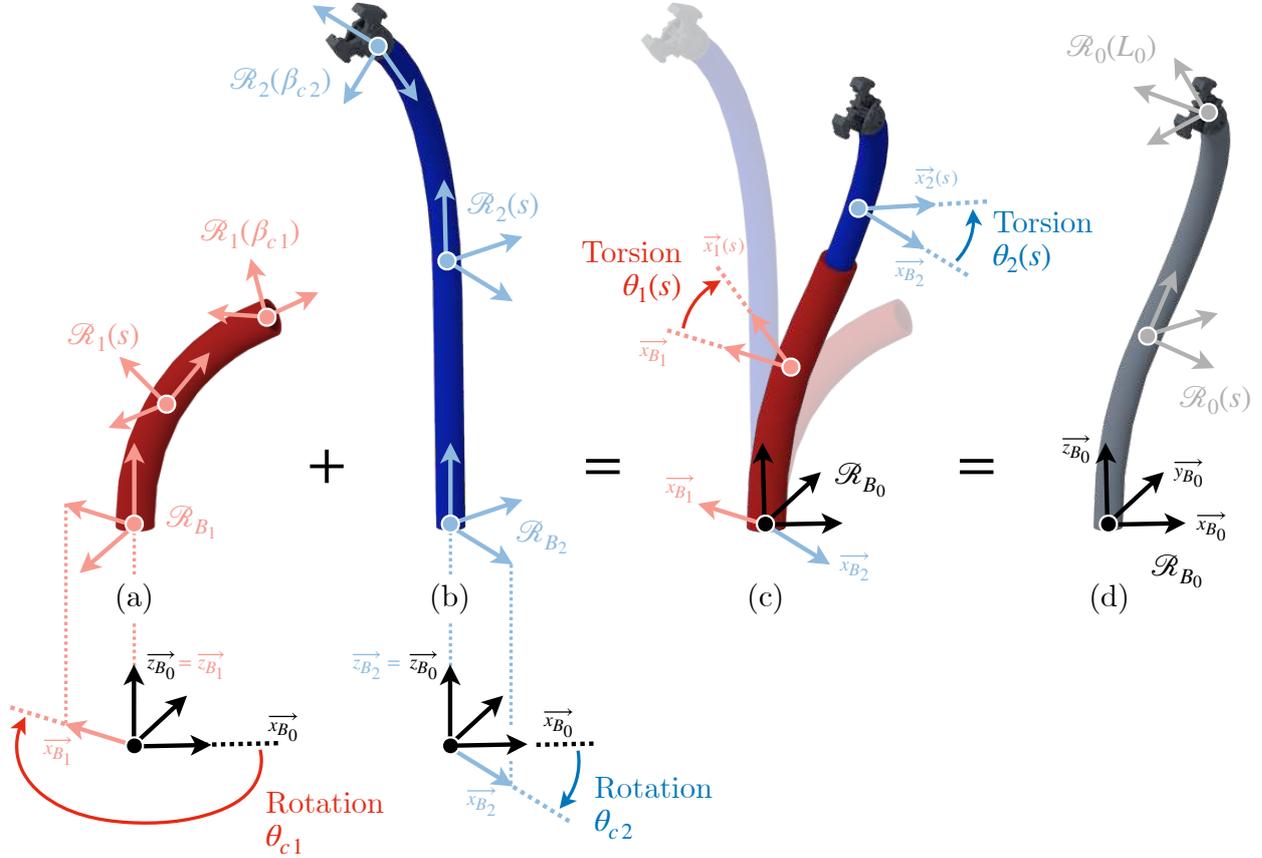


FIGURE 2.8 – Représentation des repères flottants et des angles de torsion pour un RTC composé de 2 tubes.

Dans la suite de la modélisation, nous proposerons des équations permettant entre autres de calculer l'angle de torsion des tubes en fonction de leur actionnement, de leurs propriétés matérielles et des efforts appliqués sur le robot.

2.2.3 Interaction entre les tubes

On voit désormais apparaître les variables d'actionnement θ_{ci} et β_{ci} respectivement dans les équations (2.29) et (2.32). L'objectif est alors de modéliser les déformations élastiques générées par les interactions entre les tubes. Il est admis ici que tous les tubes, une fois assemblés concentriquement, partagent tous la même forme (et donc la même courbure) qui n'est autre que la forme globale du robot. Cette hypothèse de modélisation équivaut à négliger le jeu mécanique entre les tubes. Les composantes radiales des courbures $u_i(s)|_{x,y}$ et $u_0(s)|_{x,y}$ sont les mêmes à une rotation près des repères correspondants, autour de $\vec{z}_i(s)$. La composante axiale de la courbure $u_i(s)|_z$ est quant à elle intrinsèquement liée à la torsion $\theta_i(s)$ puisque $u_i(s)|_z$ équivaut à la variation de torsion par rapport à

s. Ainsi cette hypothèse simplificatrice peut être exploitée pour obtenir les deux égalités suivantes :

$$u_i(s)|_{x,y} = \left(Rot_z(\theta_i(s))^T u_0(s) \right)|_{x,y} \quad (2.36a)$$

$$u_i(s)|_z = \dot{\theta}_i(s) \quad (2.36b)$$

En utilisant le modèle de force-déformation (2.11), il est alors possible de faire le lien entre la courbure $u_i(s)$ et le moment interne $m_i(s)$ généré par la déformation du tube i :

$$m_i(s) = K_i(s) (u_i(s) - u_i^*(s)) \quad (2.37)$$

où $K_i(s) \in \mathbb{R}^{3 \times 3}$ est la matrice de raideur en flexion-torsion de la section transverse du tube i , située à l'abscisse s .

D'après les hypothèses utilisées dans la sous-section 2.2.2, la matrice $K_i(s)$ est considérée constante sur toute la longueur du tube i . On considère également que tous les tubes sont composés du même matériau homogène et isotrope, défini par son module d'Young E et son coefficient de Poisson ν . Il est alors possible d'explicitier la matrice $K_i(s)$ comme suit :

$$K_i(s) = K_i = \begin{bmatrix} kx_i & 0 & 0 \\ 0 & kx_i & 0 \\ 0 & 0 & kz_i \end{bmatrix} = \begin{bmatrix} EI_i & 0 & 0 \\ 0 & EI_i & 0 \\ 0 & 0 & GJ_i \end{bmatrix} \quad (2.38)$$

où I_i est le moment quadratique de la section transverse du tube selon les directions $\vec{x}_i(s)$ et $\vec{y}_i(s)$, $J_i(s)$ est le moment polaire selon $\vec{z}_i(s)$ et G indique le module de cisaillement :

$$I_i = \pi \frac{R_i^4 - r_i^4}{4} \quad (2.39a)$$

$$J_i = 2I_i \quad (2.39b)$$

$$G = \frac{E}{2(\nu + 1)} \quad (2.39c)$$

$$kz_i = \nu kx_i \quad (2.39d)$$

Les efforts internes $m_i(s)$ et $n_i(s)$ peuvent également se propager le long des tubes comme décrit dans la sous-section 2.1.4. Les équations de Cosserat (2.18) restent donc valables pour le tube i , à la différence que les efforts $\tau_0(s)$ et $f_0(s)$ sont appliqués sur le robot et non sur les tubes individuellement. En retirant les termes correspondants aux efforts externes, on obtient les égalités suivantes :

$$\dot{m}_i(s) = \hat{u}_i(s)m_i(s) - \hat{v}_i(s)n_i(s) \quad (2.40a)$$

$$\dot{n}_i(s) = \hat{u}_i(s)n_i(s) \quad (2.40b)$$

où il est rappelé que, de manière analogue à (2.10), on a :

$$v_i(s) = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \quad (2.41)$$

Les efforts internes $m_0(s)$ et $n_0(s)$ du robot peuvent alors être respectivement définis comme les sommes des efforts internes $m_i(s)$ et $n_i(s)$ de chaque tube i . En prenant en compte la torsion entre le repère $\mathcal{R}_i(s)$ et le repère $\mathcal{R}_0(s)$, on obtient :

$$m_0(s) = \sum_{i \in N_T(s)} Rot_z(\theta_i(s)) m_i(s) \quad (2.42a)$$

$$n_0(s) = \sum_{i \in N_T(s)} Rot_z(\theta_i(s)) n_i(s) \quad (2.42b)$$

2.2.4 Des hypothèses au modèle différentiel

Dans la sous-section 2.2.3, des modélisations ont été formulées, permettant ainsi de disposer de relations mathématiques spécifiques à notre représentation des RTC. L'objectif est alors d'exploiter ces équations afin de compléter les équations (2.6) et (2.18) et d'obtenir une équation différentielle globale de la forme (2.19). Pour cela, il faut tout d'abord expliciter la torsion des tubes. La relation (2.36b) indique que la torsion $\theta_i(s)$ peut être décrite à partir de la courbure axiale $u_i(s)|_z$, pour laquelle nous ne disposons pas encore de description. Par dérivation de l'équation (2.37), on obtient l'égalité suivante :

$$\dot{m}_i(s) = \dot{K}_i(u_i(s) - u_i^*(s)) + K_i(\dot{u}_i(s) - \dot{u}_i^*(s)) \quad (2.43)$$

Puisque la matrice de raideur K_i est supposée constante, cette égalité peut être simplifiée sous la forme :

$$\dot{m}_i(s) = K_i(\dot{u}_i(s) - \dot{u}_i^*(s)) \quad (2.44)$$

On voit alors apparaître la forme différentielle du moment interne $m_i(s)$, ce qui permet d'injecter l'équation (2.40a) :

$$\hat{u}_i(s)m_i(s) - \hat{v}_i(s)n_i(s) = K_i(\dot{u}_i(s) - \dot{u}_i^*(s)) \quad (2.45)$$

Afin d'éliminer totalement la variable $m_i(s)$ de cette égalité, il est possible de réinjecter l'équation (2.37) pour former l'égalité suivante :

$$\hat{u}_i(s)K_i(u_i(s) - u_i^*(s)) - \hat{v}_i(s)n_i(s) = K_i(\dot{u}_i(s) - \dot{u}_i^*(s)) \quad (2.46)$$

Sachant que $v_i(s) = [0, 0, 1]^T$, on remarque que $(\hat{v}_i(s)n_i(s))|_z = 0$. Ainsi, en projetant l'égalité (2.46) sur $\vec{z}_i(s)$, il est possible d'éliminer également la variable $n_i(s)$:

$$kx_i \left(u_i(s)|_x (u_i(s)|_y - u_i^*(s)|_y) - u_i(s)|_y (u_i(s)|_x - u_i^*(s)|_x) \right) = kz_i \left(\dot{u}_i(s)|_z - \dot{u}_i^*(s)|_z \right) \quad (2.47)$$

En se référant au paramétrage des tubes, présenté dans la sous-section 2.2.2, les égalités (2.32) et (2.33) nous indiquent que le tube i n'a pas de pré-courbure en torsion. Par conséquent, nous disposons de la simplification $u_i^*(s)|_z = 0$ et $\dot{u}_i^*(s)|_z = 0$. En utilisant cette dernière dans l'équation (2.46) et en ré-organisant les variables, nous obtenons l'expression différentielle (2.48) pour décrire complètement la courbure en torsion :

$$\dot{u}_i(s)|_z = \frac{kx_i}{kz_i} (u_i(s)|_{x,y})^T \begin{bmatrix} u_i^*(s)|_y \\ -u_i^*(s)|_x \end{bmatrix} \quad (2.48)$$

Un élément encore manquant dans cette modélisation est une relation explicite entre la pré-courbure des tubes et la forme générale du robot, relation qui doit également inclure les efforts appliqués sur le robot. Un bon point de départ consiste à substituer l'expression $m_i(s)$ dans l'égalité (2.42a) par son expression donnée par le modèle force-déformation (2.37) :

$$m_0(s) = \sum_{i \in N_T(s)} Rot_z(\theta_i(s)) K_i (u_i(s) - u_i^*(s)) \quad (2.49)$$

Dès lors, il est possible de restreindre cette égalité aux axes $\vec{x}_0(s)$ et $\vec{y}_0(s)$, permettant ainsi d'injecter l'égalité (2.36a). Cette substitution permet d'introduire l'expression de la courbure radiale du robot $u_0(s)|_{x,y}$:

$$m_0(s)|_{x,y} = \left(\sum_{i \in N_T(s)} Rot_z(\theta_i(s)) K_i Rot_z(\theta_i(s))^T u_0(s) \right) \Big|_{x,y} - \left(\sum_{i \in N_T(s)} Rot_z(\theta_i(s)) K_i u_i^*(s) \right) \Big|_{x,y} \quad (2.50)$$

Puisque la matrice K_i et l'opérateur $Rot_z(\bullet)$ vérifient la relation (2.51), l'équation (2.50) peut être simplifiée pour obtenir (2.52).

$$Rot_z(\theta_i(s)) K_i Rot_z(\theta_i(s))^T = K_i \quad (2.51)$$

$$m_0(s)|_{x,y} = \left(\left(\sum_{i \in N_T(s)} K_i \right) u_0(s) \right) \Big|_{x,y} - \left(\sum_{i \in N_T(s)} Rot_z(\theta_i(s)) K_i u_i^*(s) \right) \Big|_{x,y} \quad (2.52)$$

L'égalité (2.52) propose une expression explicite directe (c'est-à-dire non-différentielle) reliant la courbure du robot et les courbures initiales des tubes. Sachant que le repère de Bishop impose $u_0(s)|_z = 0$, en isolant le terme $u_0(s)|_{x,y}$ nous obtenons la dernière relation manquante pour permettre une description complète du robot :

$$u_0(s)|_{x,y} = \left(\left(\sum_{i \in N_T(s)} K_i \right)^{-1} \left(\sum_{i \in N_T(s)} Rot_z(\theta_i(s)) K_i u_i^*(s) + m_0(s) \right) \right) \Big|_{x,y} \quad (2.53)$$

Les équations (2.6), (2.18), (2.36b) et (2.48) peuvent alors être regroupées pour obtenir un vecteur d'état $y(s)$ (2.54), régi par l'équation différentielle (2.55) et le système d'équations directes (2.56).

$$y(s) = \begin{bmatrix} \theta(s) \\ u(s)|_z \\ m_0(s) \\ n_0(s) \\ R_0^\vee(s) \\ p_0(s) \end{bmatrix} = \begin{bmatrix} \theta_1(s) \\ \vdots \\ \theta_{N_T}(s) \\ u_1(s)|_z \\ \vdots \\ u_{N_T}(s)|_z \\ m_0(s) \\ n_0(s) \\ R_0^\vee(s) \\ p_0(s) \end{bmatrix} \in \mathbb{R}^{2N_T+12} \quad (2.54)$$

$$\dot{\theta}_i(s) = u_i(s)|_z \quad (2.55a)$$

$$\dot{u}_i(s)|_z = \frac{kx_i}{kz_i} (u_i(s)|_{x,y})^T \begin{bmatrix} u_i^*(s)|_y \\ -u_i^*(s)|_x \end{bmatrix} \quad (2.55b)$$

$$\dot{m}_0(s) = \hat{u}_0(s)m_0(s) - \hat{v}_0(s)n_0(s) - R_0(s)\dot{\tau}_0(s) \quad (2.55c)$$

$$\dot{n}_0(s) = \hat{u}_0(s)n_0(s) - R_0(s)\dot{f}_0(s) \quad (2.55d)$$

$$\dot{R}_0(s) = R_0(s)\hat{u}_0(s) \quad (2.55e)$$

$$\dot{p}_0(s) = R_0(s)v_0(s) \quad (2.55f)$$

$$u_0(s)|_{x,y} = \left(\left(\sum_{i \in N_T(s)} K_i \right)^{-1} \left(\sum_{i \in N_T(s)} Rot_z(\theta_i(s)) K_i u_i^*(s) \right) + m_0(s) \right) \Big|_{x,y} \quad (2.56a)$$

$$u_0(s)|_z = 0 \quad (2.56b)$$

$$u_i(s)|_{x,y} = \left(Rot_z(\theta_{iT}(s))^T u_0(s) \right) \Big|_{x,y} \quad (2.56c)$$

$$u_i^*(s) = Rot_z(\theta_{ci}) \begin{bmatrix} \phi_i(s) \\ 0 \\ 0 \end{bmatrix} \quad (2.56d)$$

$$\phi_i(s) = \begin{cases} \frac{1}{R_{ci}} & \text{si } \max(0, \beta_{ci} - L_{ci}) \leq s \leq \beta_{ci} \\ 0 & \text{sinon} \end{cases} \quad (2.56e)$$

2.2.5 Conditions aux limites et résolution

Il a été présenté dans la [sous-section 2.1.5](#) que la résolution du modèle équivaut à l'intégration de l'équation différentielle (2.19). Cette intégration peut être directe ou indirecte en fonction des conditions aux limites connues. Dans notre cas, il est important de préciser que toutes les variables du vecteur d'état $y(s)$ n'ont pas toutes le même intervalle

d'intégration. La torsion $\theta_i(s)$ et la courbure axiale $u_i(s)|_z$, étant des variables propres à chaque tube i , ne sont définies que sur l'intervalle $s \in [0_i, \beta_{ci}]$. Les variables $m_0(s)$, $n_0(s)$, $R_0(s)$ et $p_0(s)$ sont quant à elles attachées au robot $i = 0$ et sont donc définies sur l'intégralité de l'intervalle d'intégration $[0_0, L_0]$. En ce qui concerne les conditions aux limites connues, illustrées dans la figure 2.9, elles sont réparties de part et d'autre des intervalles d'intégration.

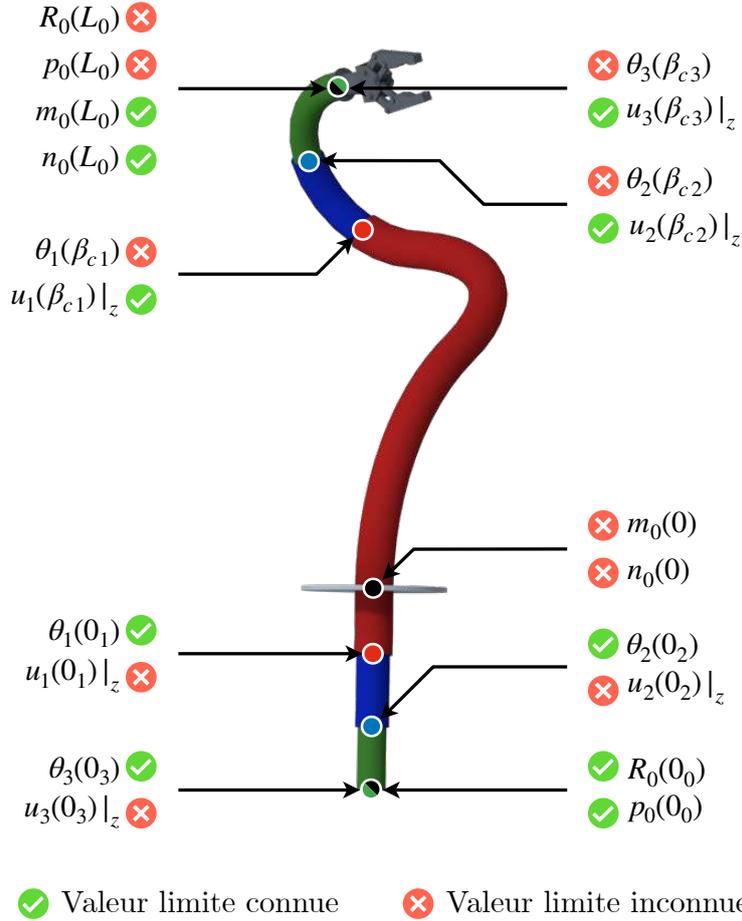


FIGURE 2.9 – Représentation des conditions aux limites connues et inconnues dans le cas d'un RTC composé de 3 tubes.

Puisqu'il est admis que les tubes sont contraints à rester rectilignes dans le mécanisme d'actionnement, la position du point $p_0(s = 0_0)$ est obtenue directement à partir des longueurs et du déploiement des tubes :

$$p_0(s = 0_0) = [0, 0, \beta_{c_{N_T}} - L_{N_T}]^T \quad (2.57)$$

Sachant que le repère de Bishop $\mathcal{R}_0(s)$ n'admet pas de torsion, l'orientation de $\mathcal{R}_0(s = 0_0)$ est identique à celle de \mathcal{R}_{B_0} . Ainsi nous pouvons écrire l'égalité suivante :

$$R_0(s = 0_0) = I_{3 \times 3} \quad (2.58)$$

Pour permettre la translation et la rotation des tubes, la base des tubes nécessite d'être fixée au mécanisme d'actionnement. Sachant que la torsion est définie par rapport au repère $\mathcal{R}_i(s)$ (et non par rapport à $\mathcal{R}_0(s)$), la valeur initiale de la torsion est nulle (et n'est pas égale à l'angle de rotation θ_{ci}) :

$$\theta_i(s = 0_i) = 0 \quad (2.59)$$

D'après la première loi de Newton, si le robot est statique alors les efforts internes et externes appliqués se compensent. En appliquant cette équilibre au bout du robot (qui est également le bout du tube $i = N_T$), il apparaît alors que les efforts internes et externes se compensent en $s = L_0$:

$$m_0(s = L_0) = R_0(L_0)^T \tau_0(L_0) \quad (2.60a)$$

$$n_0(s = L_0) = R_0(L_0)^T f_0(L_0) \quad (2.60b)$$

D'après cette même loi, les efforts doivent se compenser aux extrémités des tubes. On distingue alors deux cas : les tubes intermédiaires $i \neq N_T$ et le tube le plus long $i = N_T$. Pour les tubes $i \neq N_T$, le moment interne restreint à la torsion $m_i(\beta_{ci})|_z$ est nécessairement nulle, puisque le bout du tube est libre en torsion. En exploitant la relation force-déformation (2.37), cette contrainte est équivalente à :

$$\forall i \neq N_T, u_i(s = \beta_{ci})|_z = 0 \quad (2.61)$$

Une réflexion analogue peut être développée pour le tube le plus long $i = N_T$ à la différence que, dans ce cas, le moment externe $\tau_0(L_0)$ est appliqué, ce qui équivaut à :

$$u_{N_T}(s = \beta_{cN_T})|_z = \frac{[0 \ 0 \ 1]}{kz_{N_T}} R_{N_T}(\beta_{cN_T})^T \tau_0(L_0) \quad (2.62)$$

où $R_{N_T}(\beta_{cN_T})$ est calculé grâce à la relation :

$$R_i(s) = R_0(s) Rot_z(\theta_i(s)) \quad (2.63)$$

On dispose alors d'autant de conditions aux limites que de variables d'intégration : $2 N_T + 12$ dans chaque cas. Cependant, ces conditions aux limites connues sont réparties en deux groupes : $N_T + 6$ sont connues aux extrémités $s \in \{0_i, 0_0\}$ (2.57)-(2.59) tandis que $N_T + 6$ sont connues aux extrémités $s \in \{\beta_{ci}, L_0\}$ (2.60)-(2.62). Dans cette modélisation, il n'y a pas de sens d'intégration à privilégier puisque dans les deux sens, il y aura le même nombre de conditions initiales inconnues à optimiser grâce à la MT. Il est donc choisi arbitrairement d'intégrer les équations (2.55) dans le sens $0_0 \rightarrow L_0$. Le choix du sens d'intégration permet alors de définir le sous-vecteur d'état $y_u(s)$ (2.64), son résidu correspondant b (2.65) ainsi que de préciser les dimensions $U = B = N_T + 6$. En utilisant d'autres approches de modélisation, il est possible que $U \neq B$: dans ce cas on privilégiera le sens d'intégration pour lequel U est le plus petit afin de simplifier le problème d'optimisation.

$$y_u(s) = \begin{bmatrix} u(s)|_z \\ m_0(s) \\ n_0(s) \end{bmatrix} = \begin{bmatrix} u_1(s)|_z \\ \vdots \\ u_{N_T}(s)|_z \\ m_0(s) \\ n_0(s) \end{bmatrix} \in \mathbb{R}^U \quad (2.64)$$

$$\hat{b} = \begin{bmatrix} u_1(\beta_{c1})|_z \\ \vdots \\ u_{N_T-1}(\beta_{cN_T-1})|_z \\ u_{N_T}(L_0)|_z - \frac{[0 \ 0 \ 1]}{kz_{N_T}} R_{N_T}(L_0)^T \tau_0(L_0) \\ m_0(L_0) - R_0(L_0)^T \tau_0(L_0) \\ n_0(L_0) - R_0(L_0)^T f_0(L_0) \end{bmatrix} \in \mathbb{R}^B \quad (2.65)$$

Comme énoncé dans la [sous-section 2.1.5](#), il est nécessaire d'évaluer la jacobienne d'optimisation $\frac{\partial b}{\partial y_u(0)}$ pour utiliser la MT grâce à un algorithme de Levenberg-Marquardt. Cette matrice peut être estimée numériquement par différences finies ou calculée littéralement grâce à la méthode MPDBN, présentée dans le [chapitre 3](#).

2.2.6 Implémentation du modèle cinétostatique

Pour rappel, le modèle de RTC présenté jusque là prend en entrée les paramètres structuraux du robot Ω_{rob} ainsi que les données de commande et les efforts externes Ω_{ctrl} afin de calculer la forme complète $p_0(s)$ du RTC (2.22). Calculer ce modèle revient en pratique à résoudre un PVL construit sur le vecteur d'état (2.54) régit par les équations différentielles (2.55) et directes (2.56). La MT va itérativement affiner la valeur de $y_u(0)$ (2.64) afin de faire converger le résidu (2.65) vers $0_{B \times 1}$, à l'aide d'un algorithme de Levenberg-Marquardt. L'algorithme de résolution du modèle est synthétisé en annexe par la [figure 5.1](#).

Les paramètres de la MT peuvent avoir un impact important sur l'efficacité de la résolution du PVL. Dans cette étude, nous avons implémenté ce modèle dans Matlab R2023a [[MathWorks, 2023](#)] et les PVL sont résolus grâce à un algorithme de Levenberg-Marquardt exécuté grâce au solveur *fsolve* de Matlab. Grâce à la MPDBN présentée dans le [chapitre 3](#), la dérivée $\frac{\partial b}{\partial y_u(0)}$ est calculée efficacement et est fournie au solveur comme jacobienne d'optimisation de la fonction objectif. Il reste toutefois possible d'approximer $\frac{\partial b}{\partial y_u(0)}$ par différences finies mais cela va augmenter le temps nécessaire à la résolution du PVL.

Le seuil de convergence de l'algorithme $\epsilon = 10^{-20}$ a été choisi de manière à utiliser au maximum la précision relative sur les variables numériques de (10^{-16} pour Matlab). Comme toutes les équations différentielles sont intégrées à l'aide d'un schéma du premier

ordre, une discrétisation plus fine entraîne une augmentation du nombre d'étapes d'intégration. Par conséquent, nous avons utilisé une discrétisation variable par morceaux, notée $\Delta(s)$, grossière tout au long du robot et plus fine près des points de discontinuité $s \in \{0_i, \max(0, \beta_{ci} - L_{ci}), \beta_{ci}\}$.

L'implémentation de ce modèle a notamment permis de simuler la configuration **RTC1-C1** pour un RTC composé de 3 tubes. La [figure 2.10](#) illustre les résultats obtenus grâce à ce modèle lorsque le RTC subi ou non des forces d'interaction.

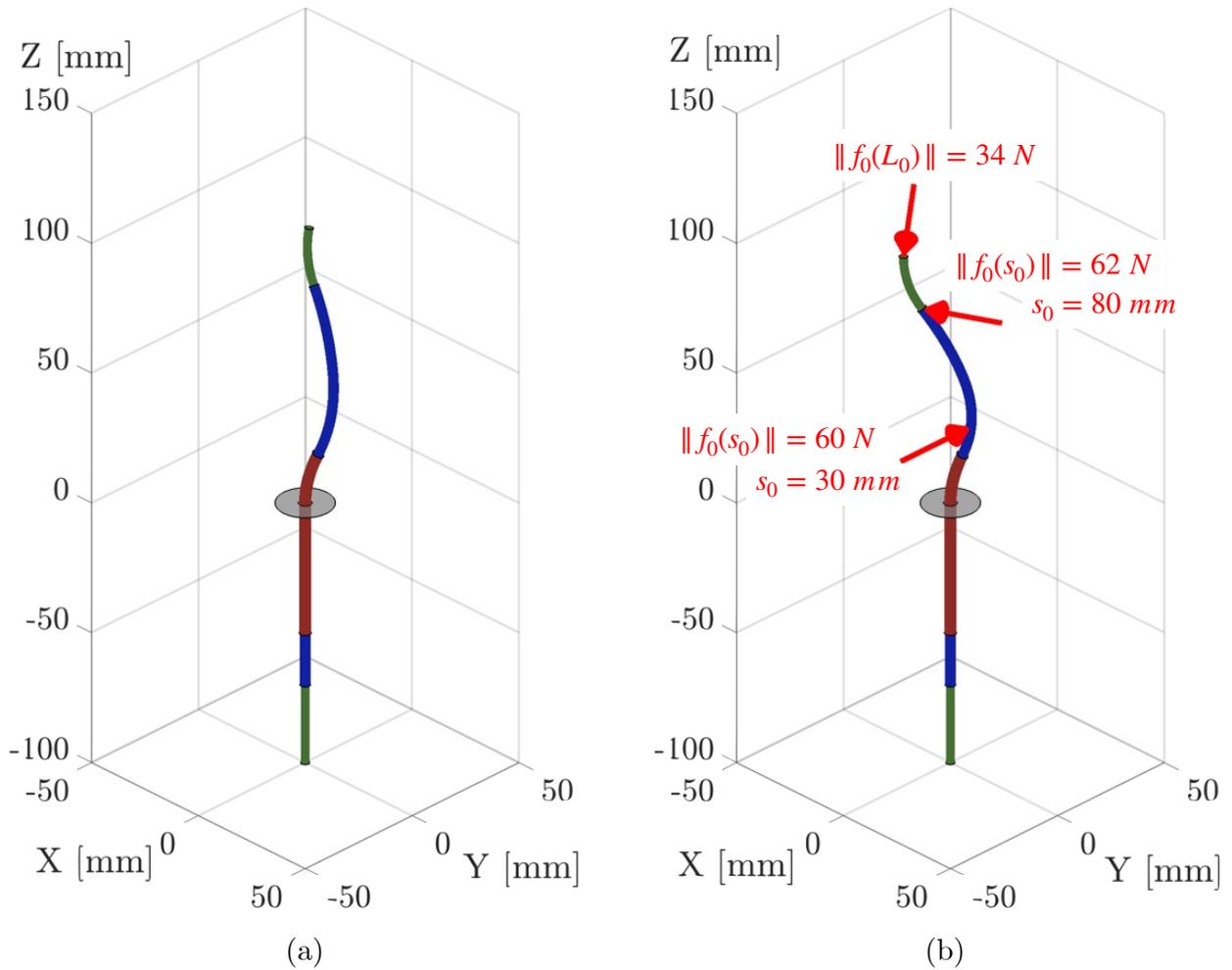


FIGURE 2.10 – Visualisation de la forme obtenue pour la configuration **RTC1-C1** à l'issue de la résolution du modèle cinéto-statique. Deux scénarios ont été simulés : lorsque le RTC ne subit aucune force d'interaction (a) et lorsque ce dernier subit les forces $\|f_0(s_0 = 30)\| = 60 \text{ N}$, $\|f_0(s_0 = 80)\| = 62 \text{ N}$ et $\|f_0(L_0 = 100)\| = 34 \text{ N}$ (b).

Lors des résolutions des PVL, nous avons observé que la MT nécessite environ $N_{ite} = 20$ itérations pour atteindre le seuil de convergence $\|b\| \leq \epsilon$. Cette valeur sera utilisée dans la [sous-section 3.2.1](#).

2.3 Conclusions

Nous avons vu au cours de ce chapitre comment construire un modèle cinétostatique analytique pour les robots continus ainsi que son implémentation pratique pour les RTC. Les équations de Cosserat [Cosserat and Cosserat, 1909], utilisées dans ce modèle, permettent d'inclure les efforts d'interaction dans le modèle de la forme. On rappelle que la modélisation présentée n'est pas une contribution en tant que telle mais est une réinterprétation de travaux existants, notamment ceux de [Lock et al., 2010] et ceux de [Rucker et al., 2010].

Les essais expérimentaux réalisés par [Lock et al., 2010] sur un RTC composé de 2 tubes ont montré que la position du bout du robot peut être estimée avec une erreur moyenne de moins de 4 mm lorsqu'une force est appliquée au bout du robot. Les travaux réalisés par [Rucker et al., 2010] ont eux montré que la position du bout du robot peut être modélisée avec une erreur moyenne de 2.89 mm lorsqu'aucune force n'est appliquée sur le robot et de 2.92 mm lorsqu'une force est uniformément appliquée sur toute la longueur du robot. Il est toutefois important de garder à l'esprit que ces erreurs d'estimation peuvent dépendre de la longueur totale du robot : plus la distance de déploiement augmente, plus les erreurs s'accroissent, ce qui entraîne une augmentation de l'erreur sur la position de l'extrémité du robot.

Si on utilise uniquement le modèle pour commander un RTC, c'est-à-dire si on le commande en boucle ouverte, une erreur de position de plus de 2 mm semble trop grande pour envisager des interventions chirurgicales nécessitant parfois une précision de positionnement submillimétrique comme c'est le cas pour les micro-chirurgies des yeux [Riviere et al., 1997]. Grâce à une commande en boucle fermée exploitant une mesure de la position du bout d'un RTC, [Boyer et al., 2024] ont réussi à réduire cette erreur à environ 0.3 mm, ouvrant ainsi de nouvelles perspectives pour les applications chirurgicales.

Comme pour tout modèle théorique, il serait possible de réduire davantage l'erreur de modélisation de la forme en levant des hypothèses simplificatrices. D'autres modèles ont été développés en ce sens : [Lock and Dupont, 2011] et [Ha et al., 2018] ont respectivement proposé d'inclure le frottement et le jeu mécanique entre les tubes. Les modèles obtenus sont plus réalistes mais également considérablement plus complexes. Puisque le réalisme du modèle cinétostatique présenté semble d'ores et déjà raisonnable pour de nombreuses applications chirurgicales, nous allons nous intéresser dans le chapitre suivant à la modélisation d'une autre propriété mécanique : la compliance.

Chapitre 3

Modélisation cinématique et compliance

Nous avons vu dans le [chapitre 2](#) comment modéliser la forme d'un robot continu ainsi que les équations utilisées dans le cas d'un RTC. En pratique, l'utilisation du modèle cinéto-statique direct est limitée car de nombreuses applications nécessitent l'utilisation du modèle cinéto-statique inverse. Par exemple lorsque l'on souhaite déployer le robot de sorte à ce qu'il atteigne une cible clinique donnée, le chirurgien va le plus souvent définir une trajectoire que le robot devra suivre [[Rucker et al., 2013](#)]. Prédire la forme du robot pour un actionnement donné n'est pas suffisant, il faut dans ce cas prédire l'actionnement du robot permettant d'obtenir une forme donnée. Sachant que le modèle cinéto-statique inverse peut être difficile à formuler, une approche alternative, couramment utilisée en robotique, consiste à utiliser la matrice jacobienne du robot, apparaissant lorsque l'on dérive le modèle cinéto-statique direct, pour inverser le problème numériquement. Le calcul des dérivées du modèle ne se substitue pas au modèle cinéto-statique mais le complète car ces dérivées ne sont valables que localement et pour une forme donnée [[Murray et al., 1994](#)]. Dans la première partie de ce chapitre, la [section 3.1](#) présentera les différentes dérivées définies pour les robots continus et notamment celles associées au concept de compliance. Après discussion, nous proposerons à la fin de la section une nouvelle définition de la compliance, permettant de décrire de manière plus globale les propriétés mécaniques des robots continus. Au cours de la [section 3.2](#), nous nous attarderons sur les méthodes permettant de calculer une telle compliance, ce qui nous amènera à proposer une nouvelle méthode de calcul spécifiquement développée pour cette nouvelle définition. Dans la dernière partie, [section 3.3](#), cette nouvelle méthode sera appliquée aux RTC et validée grâce à des simulations numériques. Les contributions principales de ce chapitre sont présentées dans la [sous-section 3.1.2](#) et la [sous-section 3.2.4](#).

Les nouvelles variables introduites dans ce chapitre sont disponibles en annexes dans le [tableau 5.11](#).

3.1 Définition de la compliance

La dérivation des modèles de robots continus est un sujet de recherche déjà largement étudié et qui a bénéficié de travaux développés tout d’abord pour la robotique rigide. L’état de l’art [sous-section 3.1.1](#) présente les techniques de dérivation employées dans le cas des robots continus. Une discussion sur la compliance des robots continus fera apparaître une limite majeure des définitions existantes. Dans la [sous-section 3.1.2](#), nous proposerons une nouvelle définition, plus générale que celles actuelles, pour représenter le concept de compliance.

3.1.1 Introduction

En analyse vectorielle, il est courant d’utiliser l’opérateur jacobien pour représenter la matrice des dérivées partielles d’une fonction. La robotique continue définit *la jacobienne articulaire d’un robot*, ou plus simplement *la jacobienne*, comme la matrice des dérivées partielles de la pose $T_0^V(s)$ par rapport aux variables articulaires q . Pour de nombreuses applications, cette jacobienne est définie pour l’extrémité distale $s = L_0$ où est fixé l’effecteur du robot. Dans la littérature, la jacobienne d’un robot est usuellement notée $J(q)$ [[Cheah et al., 2004](#)] mais dans cette étude, elle sera notée $J(L_0)$ par soucis de cohérence avec les prochaines définitions présentées :

$$J(L_0) = \frac{\partial T_0^V(L_0)}{\partial q} \quad (3.1)$$

Dans le cas d’un robot continu, la tâche à réaliser par le robot, notamment dans des scénarios médicaux, peut concerner, en plus de la position de l’extrémité $s = L_0$, l’ensemble du corps flexible $s \neq L_0$. C’est le cas par exemple lors du déploiement d’un endoscope robotisé le long d’une trajectoire sûre. La définition (3.1) peut alors être étendue en considérant l’ensemble des repères $\mathcal{R}_0(s)$ et non uniquement le repère de l’effecteur $\mathcal{R}_0(s = L_0)$ [[Rucker and Webster, 2011](#)]. Cette jacobienne plus générale est définie par (3.2) et illustrée par la [figure 3.1](#).

$$\forall s \in [0, L_0], J(s) = \frac{\partial T_0^V(s)}{\partial q} \quad (3.2)$$

Le lecteur remarquera que la définition (3.2) inclut le cas (3.1) où $s = L_0$. La définition (3.2) permet alors d’envisager un suivi de trajectoire pour l’effecteur $\mathcal{R}_0(s)$ tout en assurant que le corps du robot $\mathcal{R}_0(s \neq L_0)$ ne rentre pas en contact avec des obstacles environnants [[Ataka et al., 2016](#)]. Toutefois, les objectifs de commande ou de planification des mouvements du robot ne se restreignent pas à des contraintes sur la pose. Lorsque le robot interagit avec un objet particulièrement fragile, comme lors d’une chirurgie ophthalmologique par exemple, des contraintes de compliance (ou de rigidité) peuvent être imposées au robot afin d’éviter des dommages éventuels [[Wang et al., 2023b](#)]. Grâce à leur flexibilité intrinsèque, les robots continus peuvent alors absorber les efforts en se déformant. On trouve dans la littérature une autre forme de matrice jacobienne : la matrice

3.1. Définition de la compliance

des dérivées partielles de la pose $T_0^\vee(s)$ par rapport à un effort $w_0(L_0)$ [Aloi et al., 2018]. Cette dernière, notée $C(L_0)$, est désignée comme *la matrice de compliance* pour éviter toute confusion avec la jacobienne articulaire :

$$C(L_0) = \frac{\partial T_0^\vee(L_0)}{\partial w_0(L_0)} \quad (3.3)$$

De manière analogue à la jacobienne (3.2), la définition de la matrice de compliance (3.3), au niveau de l'extrémité du robot, est assez restrictive. Sachant que les robots continus sont flexibles, la propagation des efforts déformera l'intégralité de la longueur du robot et non juste l'extrémité. La compliance peut alors être étendue en considérant l'ensemble des repères $\mathcal{R}_0(s \neq L_0)$ et non uniquement le repère $\mathcal{R}_0(s = L_0)$ [Rucker and Webster, 2011, Smoljkic et al., 2014]. Cette matrice de compliance plus générale est définie par (3.4) et est illustrée par la figure 3.2. De manière analogue à la jacobienne articulaire, le lecteur remarquera que la définition (3.4) inclut le cas (3.3) où $s = L_0$.

$$\forall s \in [0, L_0], C(s) = \frac{\partial T_0^\vee(s)}{\partial w_0(L_0)} \quad (3.4)$$

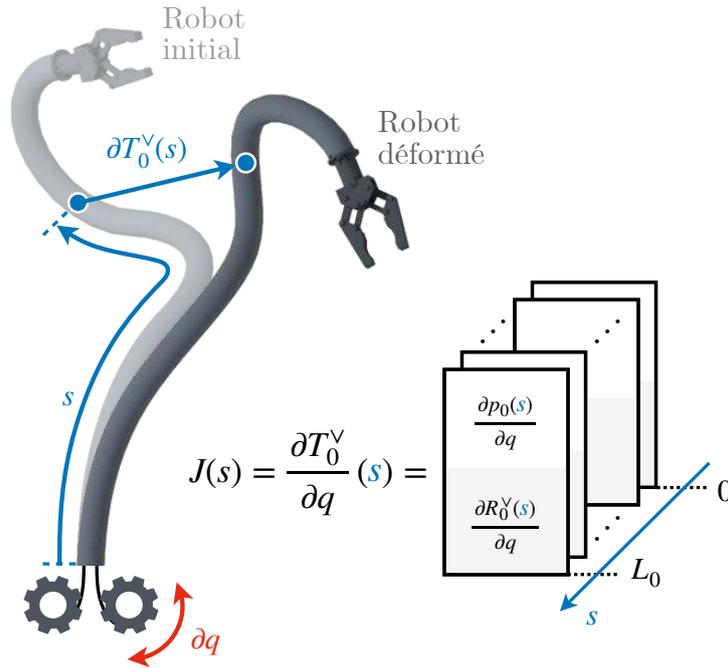


FIGURE 3.1 – Illustration et représentation matricielle de la définition de la jacobienne articulaire selon (3.2).

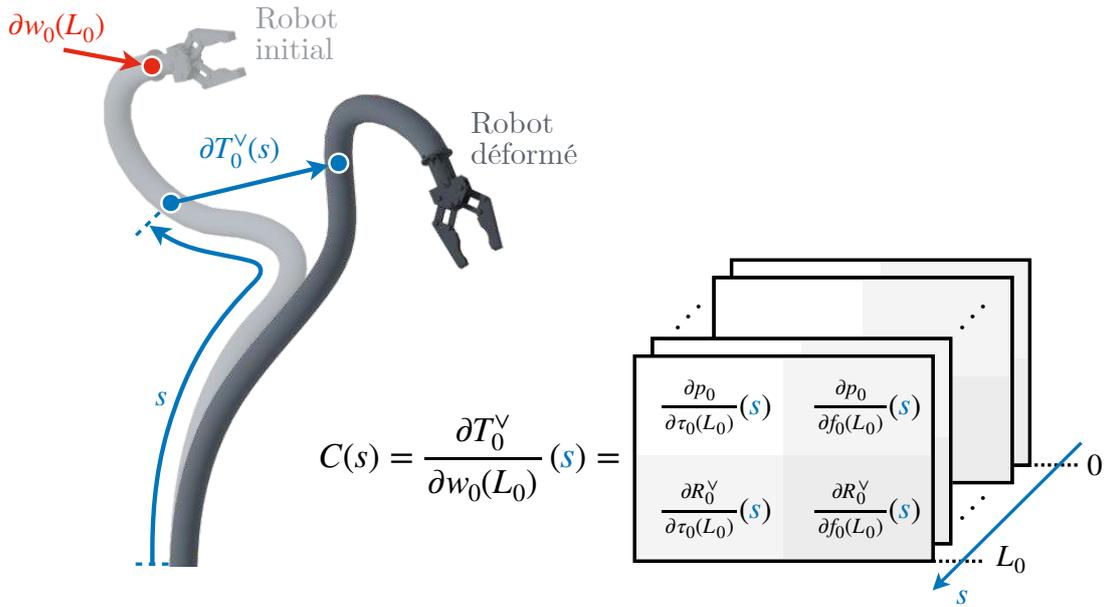


FIGURE 3.2 – Illustration et représentation de la matrice de compliance selon (3.3).

3.1.2 La matrice de compliance généralisée

La définition (3.4) de la matrice de compliance n'offre toujours pas une description exhaustive de la compliance d'un robot continu. Pour tenir compte d'une plus grande variété de situations, nous proposons une définition plus générale dans laquelle le point d'application de l'effort $w_0(s_0)$ est paramétré par son abscisse curviligne s_0 . Cette nouvelle matrice, notée $C_{s_0}(s)$, est définie par (3.5) et illustrée par la figure 3.3.

$$\forall (s, s_0) \in [0, L_0]^2, C_{s_0}(s) = \frac{\partial T_0^\vee(s)}{\partial w_0(s_0)} \quad (3.5)$$

On remarquera que la définition (3.5) inclut les cas particuliers (3.4) et (3.3). La matrice de la compliance généralisée (3.5) utilise deux paramètres indépendants : l'abscisse curviligne s d'un point quelconque du robot et l'abscisse curviligne s_0 du point d'application de l'effort $w_0(s_0)$, quelconque également. En raison du nouveau paramètre s_0 , $C_{s_0}(s)$ dispose d'une dimension de plus que $C(s)$ pour un total de 4 dimensions. On gardera le terme de *matrice* au lieu de *tenseur* par analogie aux définitions de la compliance pré-existantes. Les 4 dimensions de la matrice de compliance généralisée peuvent être organisées de la manière suivante :

- La première regroupe les 6 composantes de $T_0^\vee(s)$ qui sont dérivées ;
- La deuxième regroupe les 6 composantes de $w_0(s_0)$ par rapport auxquelles la dérivation est réalisée ;
- La troisième indique l'abscisse curviligne s considérée ;
- La quatrième indique l'abscisse curviligne s_0 considérée.

3.1. Définition de la compliance

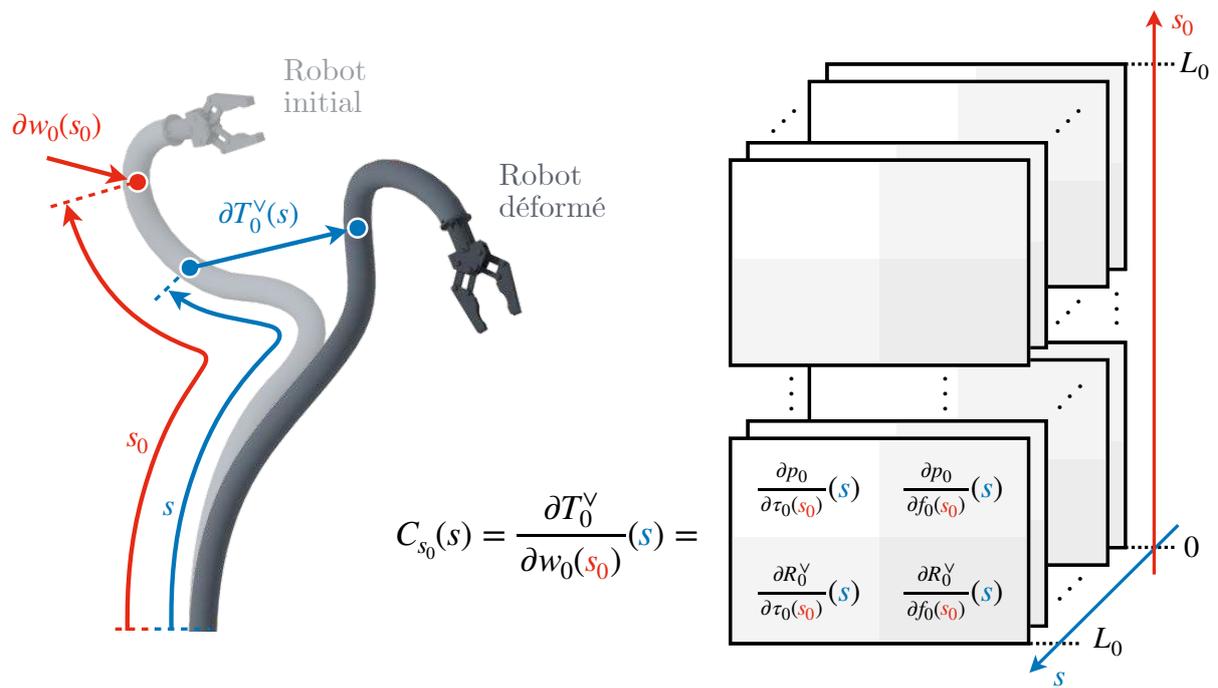


FIGURE 3.3 – Illustration graphique du concept de compliance généralisée proposé dans cette thèse (a) et de sa représentation matricielle (b).

3.2 Calcul de la compliance généralisée

Bien que la nouvelle définition (3.5) permette de représenter plus d'informations que les définitions existantes, il est nécessaire de proposer une méthode de calcul adaptée pour pouvoir l'utiliser efficacement dans diverses applications. Sachant que les définitions (3.2) et (3.4) ont d'ores et déjà été étudiées [Rucker and Webster, 2011, Khadem et al., 2018, Xu et al., 2013], un bon point de départ consiste à analyser les méthodes de calcul existantes. Les méthodes décrites seront présentées suivant leur complexité croissante : une première méthode totalement numérique sera exposée dans la sous-section 3.2.1, une seconde méthode alliant des calculs littéraux et numériques sera détaillée dans la sous-section 3.2.2 et finalement nous présenterons une méthode purement littérale dans la sous-section 3.2.3. Nous discuterons des limites de cette dernière méthode, mettant en évidence des améliorations possibles lors du calcul de la compliance. Enfin, nous détaillerons dans la sous-section 3.2.4 une nouvelle méthode intégrant ces améliorations et particulièrement adaptée pour la nouvelle définition de la compliance.

3.2.1 Calcul par accroissements finis numériques

Une méthode standard et simple d'implémentation pour estimer la matrice de compliance généralisée consiste à faire une approximation par accroissements finis sur la fonction \mathcal{M} (2.22). Cette approche, utilisée pour calculer $C(s)$ (3.4) [Khadem et al., 2018], doit être adaptée pour prendre en compte les nouveaux dérivateurs $w_0(s_0 \neq L_0)$. En introduisant la notation $\Delta\bullet$ pour indiquer une variation de la variable \bullet , la compliance généralisée $C_{s_0}(s)$ est approximée par :

$$C_{s_0}(s) \approx \frac{\mathcal{M}(s, \Omega_{rob}, \Omega_{ctrl} + \Delta w_0(s_0)) - \mathcal{M}(s, \Omega_{rob}, \Omega_{ctrl} - \Delta w_0(s_0))}{2\Delta w_0(s_0)} \quad (3.6)$$

Cette méthode, que nous désignerons sous l'acronyme MDFPVL (Méthode par Différences Finies sur le Problème d'intégration aux Valeurs Limites), peut être facilement implémentée : elle ne nécessite qu'un algorithme simple construit directement à partir de la fonction \mathcal{M} . La contrepartie à la simplicité de la MDFPVL est un temps de calcul très significatif. Comme présenté dans la sous-section 2.1.5, dans le cas d'un PVL résolu grâce à une MT, la résolution du problème nécessite des intégrations itératives du PVI jusqu'à ce que le résidu b converge vers $0_{B \times 1}$. Bien que le nombre d'itérations N_{ite} nécessaires pour faire converger la MT ne soit pas constant, nous utiliserons par la suite la valeur moyenne $N_{ite} = 20$ issue des observations de la section 2.2.

Pour obtenir une approximation complète de la matrice de compliance, l'accroissement fini (3.6) doit être calculé pour les 6 composantes de $w_0(s_0)$ ainsi que pour tous les $s_0 \in [0, L_0]$, ce qui fait un total de $6N_P$ accroissements. De plus, chaque intégration du PVI équivalent nécessite YN_P opérations auxquelles il faut ajouter B opérations pour le calcul du résidu b . Finalement, le nombre total d'opérations, noté N_{PVL}^{DF} , nécessaires pour estimer $C_{s_0}(s)$ par accroissements finis à partir de la fonction \mathcal{M} est donné par (3.7).

$$N_{PVL}^{DF} = 2 \cdot 6 \cdot N_P \cdot N_{ite} \cdot (Y N_P + B + 6 N_P) \quad (3.7a)$$

$$= N_P^2 \cdot (N_{ite} (12Y + 72)) + N_P \cdot (12 N_{ite} B) \quad (3.7b)$$

où \bullet^{DF} indique que le calcul fait appel à des différences finies et \bullet_{PVL} précise que ces dernières sont appliquées directement sur le PVL.

Il convient de préciser que l'égalité (3.7) donne une approximation du nombre d'opérations et n'est pas exacte. Cette égalité est construite à partir d'une estimation des opérations liées au processus d'intégration et ne prend pas en compte ni les opérations de calcul direct ni les opérations logicielles liées à la manipulation des différentes variables. Cependant l'égalité (3.7) permet tout de même d'avoir un ordre de grandeur du nombre d'opérations et d'identifier les pistes d'amélioration pour réduire le temps de calcul (intrinsèquement lié au nombre d'opérations).

On notera que la présence de la variable N_{ite} signifie que le nombre d'opérations N_{PVL}^{DF} n'est pas forcément constant ce qui peut complexifier l'intégration du calcul de $C_{s_0}(s)$ dans un algorithme de commande en temps réel. Sachant que le modèle géométrique \mathcal{M} est d'ores et déjà résolu lorsque l'on calcule la compliance, on peut alors se demander s'il serait possible d'éviter l'utilisation systématique de la MT pour chaque accroissement fini.

3.2.2 Calcul hybride : littéral et différences finies numériques

Une approche originale consiste à décomposer les dérivées (3.2) et (3.4) en des sommes de dérivées relatives à la forme et au résidu [Rucker and Webster, 2011]. Cette approche permet de calculer la jacobienne et la matrice compliance sans avoir à résoudre à nouveau le modèle géométrique complet \mathcal{M} et ainsi éviter l'utilisation de la MT. Cette méthode a été présentée dans [Rucker and Webster, 2011] dans le cadre du calcul de la compliance telle qu'exprimée dans l'équation (3.4). L'objectif étant de calculer la compliance généralisée définie par (3.5), nous allons donc présenter par la suite une adaptation de la méthode originale. Toutefois les équations relatives à la jacobienne (3.2) restent quant à elles inchangées. Une hypothèse de modélisation supplémentaire est nécessaire pour cette décomposition : on suppose qu'un changement de la pose du robot $T_0^\vee(s)$ ne peut être dû que par une variation d'actionnement q ou une variation d'efforts $w_0(s_0)$, et que celles-ci sont indépendantes. Ainsi, nous ne considérons pas de variations de formes éventuellement générées par un changement de température ou par une évolution des propriétés matérielles du robot. Cette hypothèse se traduit par l'égalité suivante :

$$\left. \frac{dT_0^\vee(s)}{dt} \right|_{\odot} = J(s) \frac{dq}{dt} + C_{s_0}(s) \frac{dw_0(s_0)}{dt} \quad (3.8)$$

où la notation $\bullet|_{\odot}$ précise que la contrainte $b = 0_{B \times 1}$ est garantie malgré la variation de forme et l'opérateur $\frac{d\bullet}{dt}$ indique une dérivation par rapport au temps t .

Pour simplifier la suite des équations, nous introduisons la variable muette χ qui permet de décrire en une fois des expressions appliquées à $y_u(0)$, q et $w_0(s_0)$:

$$\chi \in \{y_u(0), q, w_0(s_0)\} \quad (3.9)$$

Supposons alors que les conditions initiales $y_u(0)$ soient indépendantes de l'actionnement q et des efforts $w_0(s_0)$ (cela équivaut à ne plus garantir la contrainte $b = 0_{B \times 1}$). Dans ce cas, un changement de la forme peut également être dû à un changement de conditions initiales $y_u(0)$, ce qui nous donne la nouvelle égalité :

$$\left. \frac{dT_0^\vee(s)}{dt} \right|_{\otimes} = E_q(s) \frac{dq}{dt} + E_{w_0(s_0)}(s) \frac{dw_0(s_0)}{dt} + E_{y_u(0)}(s) \frac{dy_u(0)}{dt} \quad (3.10)$$

où la notation $\bullet|_{\otimes}$ précise que la contrainte $b = 0_{B \times 1}$ n'est pas forcément respectée. De plus, les variables E_χ représentent les dérivées de $T_0^\vee(s)$ par rapport à χ , sans contrainte sur le résidu :

$$E_\chi(s) = \left. \frac{\partial T_0^\vee(s)}{\partial \chi} \right|_{\otimes} \quad (3.11)$$

Il est important de souligner que la jacobienne $J(s)$ et la dérivée $E_q(s)$ se distinguent l'une de l'autre respectivement par le respect et le non respect de la contrainte $b = 0_{B \times 1}$. On retrouve la même distinction entre $C_{s_0}(s)$ et $E_{w_0(s_0)}(s)$. De manière analogue à l'égalité (3.10), il est possible de décomposer les variations de résidu b de la manière suivante :

$$\left. \frac{db}{dt} \right|_{\otimes} = B_q \frac{dq}{dt} + B_{w_0(s_0)} \frac{dw_0(s_0)}{dt} + B_{y_u(0)} \frac{dy_u(0)}{dt} \quad (3.12)$$

où les variables B_χ représentent les dérivées du résidu b par rapport à χ :

$$B_\chi = \left. \frac{\partial b}{\partial \chi} \right|_{\otimes} \quad (3.13)$$

On remarquera dans la définition (3.13) que les variables B_χ ne sont pas paramétrées selon l'abscisse curviligne s puisque le résidu b est systématiquement défini à l'extrémité du robot $s = L_0$. On rappelle que la résolution du modèle revient à trouver les conditions initiales $y_u(0)$ telles que $b = 0_{B \times 1}$. Si cette contrainte est respectée, on peut alors ré-écrire l'égalité (3.12) pour obtenir :

$$\left(b = 0_{B \times 1} \right) \implies \left(\left. \frac{db}{dt} \right|_{\odot} = 0_{B \times 1} = B_q \frac{dq}{dt} + B_{w_0(s_0)} \frac{dw_0(s_0)}{dt} + B_{y_u(0)} \frac{dy_u(0)}{dt} \right) \quad (3.14)$$

En isolant la dérivée sur les conditions initiales et en réorganisant l'égalité (3.14), on obtient alors la nouvelle expression :

$$\left(b = 0_{B \times 1} \right) \implies \left(\left. \frac{dy_u(0)}{dt} \right|_{\odot} = -B_{y_u(0)}^\dagger \left(B_q \frac{dq}{dt} + B_{w_0(s_0)} \frac{dw_0(s_0)}{dt} \right) \right) \quad (3.15)$$

où \bullet^\dagger est l'opérateur de la pseudo-inverse de Moore–Penrose.

Lorsque l'on substitue $\frac{dy_u(0)}{dt}$ de l'égalité (3.10) par sa nouvelle expression (3.15), cela garantit alors que la contrainte $b = 0_{B \times 1}$ soit respectée. En faisant passer la notation de $\bullet|_{\otimes}$ à $\bullet|_{\odot}$, il est alors possible de ré-écrire l'égalité (3.10) :

$$\begin{aligned} \left. \frac{dT_0^\vee(s)}{dt} \right|_{\odot} &= E_q(s) \frac{dq}{dt} + E_{w_0(s_0)}(s) \frac{dw_0(s_0)}{dt} \\ &\quad - E_{y_u(0)}(s) B_{y_u(0)}^\dagger \left(B_q \frac{dq}{dt} + B_{w_0(s_0)} \frac{dw_0(s_0)}{dt} \right) \end{aligned} \quad (3.16)$$

En factorisant l'expression (3.16) par $\frac{dq}{dt}$ et $\frac{dw_0(s_0)}{dt}$, on trouve finalement l'expression suivante :

$$\begin{aligned} \left. \frac{dT_0^\vee(s)}{dt} \right|_{\odot} &= \left(E_q(s) - E_{y_u(0)}(s) B_{y_u(0)}^\dagger B_q \right) \frac{dq}{dt} \\ &\quad + \left(E_{w_0(s_0)}(s) - E_{y_u(0)}(s) B_{y_u(0)}^\dagger B_{w_0(s_0)} \right) \frac{dw_0(s_0)}{dt} \end{aligned} \quad (3.17)$$

L'égalité (3.17) permet, par identification avec l'égalité (3.8), d'obtenir des formes décomposées pour la jacobienne $J(s)$ et pour la compliance généralisée $C_{s_0}(s)$:

$$J(s) = E_q(s) - E_{y_u(0)}(s) B_{y_u(0)}^\dagger B_q \quad (3.18a)$$

$$C_{s_0}(s) = E_{w_0(s_0)}(s) - E_{y_u(0)}(s) B_{y_u(0)}^\dagger B_{w_0(s_0)} \quad (3.18b)$$

On remarque que l'expression (3.18b) est construite à partir de deux termes. Le premier terme $E_{w_0(s_0)}(s)$ est relié à la déformation directe du robot, c'est-à-dire celle générée directement par une variation d'effort $w_0(s_0)$, sans tenir compte du résidu b . Le second terme $-E_{y_u(0)}(s) B_{y_u(0)}^\dagger B_{w_0(s_0)}$ représente quant à lui la déformation indirecte de la forme : la variation d'efforts $w_0(s_0)$ va faire évoluer la valeur du résidu b , qui va donc modifier les conditions initiales $y_u(0)$ qui vont à leur tour déformer le robot. La somme de ces deux termes permet donc de décomposer la dérivation globale de la forme en une somme de dérivations directes et indirectes. Cette approche permet ainsi de calculer $C_{s_0}(s)$ sans avoir à résoudre à nouveau le PVL. Une analyse similaire peut être menée pour l'équation (3.18a) et le calcul de la jacobienne $J(s)$.

Grâce à l'équation (3.18b), on peut alors appliquer les accroissements finis pour évaluer $E_{w_0(s_0)}(s)$, $E_{y_u(0)}(s)$, $B_{y_u(0)}$ et $B_{w_0(s_0)}$ puis calculer directement $C_{s_0}(s)$. La démarche présentée consiste en quelque sorte à appliquer des accroissements finis sur le PVI, et non sur le PVL. Cette méthode hybride, alliant calculs analytiques et numériques, permet de réduire le nombre d'opérations puisque chaque variable E_x ou B_x peut être estimée en une seule intégration, et non en $N_{ite} = 20$ itérations comme pour la MDFPVL. Le nombre d'opérations nécessaires pour cette méthode, noté N_{PVI}^{DF} , suit désormais l'expression donnée par l'équation (3.19b).

$$N_{PVI}^{DF} = 2 \cdot 6 \cdot N_P \cdot (YN_P + B + 6N_P) + 2 \cdot U \cdot (YN_P + B + 6N_P) + 36N_P^2 \quad (3.19a)$$

$$= N_P^2 \cdot (12Y + 108) + N_P \cdot (12B + 2UY + 12U) + UB \quad (3.19b)$$

où \bullet_{PVI} précise que ces dernières sont appliquées sur le PVI et que la compliance est ensuite calculée grâce à l'égalité (3.18b).

Nous désignerons par la suite cette méthode hybride sous l'acronyme MDFPVI (Méthode par Différences Finies sur le Problème d'intégration aux Valeurs Initiales). L'expression (3.19) ne comporte plus la variable N_{ite} contrairement à (3.7), ce qui signifie que le nombre d'opérations pour le calcul de la compliance reste constant peu importe la configuration du robot. Bien que la MDFPVI soit plus efficace que la MDFPVL, les deux souffrent d'un même inconvénient : l'utilisation d'accroissements finis implique nécessairement des erreurs d'approximation. Étant donné que l'équation (3.18b) permet de décomposer la compliance en une somme de plusieurs dérivées, on peut alors se demander s'il ne serait pas possible de calculer de manière littérale les variables E_χ et B_χ . Cette approche assurerait un calcul théoriquement exact (d'un point de vue géométrique) et permettrait probablement de réduire le nombre d'opérations.

3.2.3 Calcul littéral par propagation de dérivées haut-niveau

En plus d'avoir présenté des décompositions pour les matrices (3.2) et (3.4), Rucker et al. ont également présenté une méthode littérale pour calculer les dérivées $E_\chi(s)$ et B_χ [Rucker and Webster, 2011]. Pour cela, il est nécessaire d'introduire une nouvelle variable, notée $V_\chi(s)$, qui représente la dérivée partielle du vecteur d'état $y(s)$ par rapport à χ :

$$V_\chi(s) = \frac{\partial y}{\partial \chi}(s) \quad (3.20)$$

En l'état, il n'est pas possible de calculer directement $V_\chi(s)$ car nous ne disposons pas d'équation analytique directe pour le vecteur d'état $y(s)$. Cependant, nous avons vu dans le chapitre 2 que l'équation différentielle (2.19) permet de calculer $y(s)$ via le calcul de sa propagation le long du robot : nous allons donc calculer $V_\chi(s)$ par propagation en construisant une équation différentielle dédiée. La méthode consiste à dériver $V_\chi(s)$ par rapport à s (en rappelant que χ et s sont indépendants), à ré-utiliser l'égalité (2.19) puis à appliquer le théorème de dérivation des fonctions composées. Le développement correspondant est donné par l'équation (3.21) :

$$\dot{V}_\chi(s) = \frac{d}{ds} \left(\frac{\partial y}{\partial \chi} \right) (s) = \frac{\partial}{\partial \chi} \left(\frac{dy}{ds} \right) (s) = \frac{\partial \mathcal{F}}{\partial \chi}(y, s) = \frac{\partial \mathcal{F}}{\partial y}(y, s) \frac{\partial y}{\partial \chi}(s) \quad (3.21)$$

On observe que l'expression de $V_\chi(s)$ ré-apparaît dans l'égalité (3.21) ce qui nous donne l'équation suivante :

$$\dot{V}_\chi(s) = \frac{\partial \mathcal{F}}{\partial y}(y, s) V_\chi(s) \quad (3.22)$$

Cette méthode de calcul est désignée sous le nom de "Méthode par Propagation de Dérivées" car l'équation (3.22) permet de calculer $V_\chi(s)$ grâce à sa propagation de long du robot. Cette dernière sera utilisée par la suite dans le calcul de B_χ , définie par (3.13).

Le calcul de B_χ a quant à lui une subtilité qu'il convient de préciser : une variation $\partial\chi$ impacte b de deux manières différentes : de manière directe si χ apparaît explicitement dans l'expression de b et de manière indirecte à travers les variations de $y(L_0)$. Pour cette raison, on peut alors décomposer (3.13) grâce au théorème de dérivation des fonctions composées :

$$B_\chi = \frac{\partial b|_{\text{''}b\text{''}}}{\partial y(L_0)} \frac{\partial y}{\partial \chi}(L_0) + \frac{\partial b|_{\text{''}b\text{''}}}{\partial \chi} \quad (3.23)$$

où la notation $\bullet|_{\text{''}b\text{''}}$ précise que c'est l'expression de b qui est dérivée et non la variable en tant que telle. On voit alors apparaître dans l'égalité (3.23) la dérivée $V_\chi(L_0)$:

$$B_\chi = \frac{\partial b|_{\text{''}b\text{''}}}{\partial y(L_0)} V_\chi(L_0) + \frac{\partial b|_{\text{''}b\text{''}}}{\partial \chi} \quad (3.24)$$

L'équation (3.24) apporte avec elle trois nouveautés majeures par rapport à la méthode originale de [Rucker and Webster, 2011] :

- La première différence porte sur le type d'équation utilisée pour B_χ . L'équation (14) de [Rucker and Webster, 2011] présente une équation différentielle sur B_χ alors qu'en pratique c'est une équation directe qu'il faut utiliser dans ce cas : c'est une erreur qui sera corrigée par l'équation (5.14) du manuscrit de thèse de l'auteur [Rucker, 2011].
- La seconde différence peut être observée lors de l'application du théorème de dérivation des fonctions composées. Les équations (14) de [Rucker and Webster, 2011] et (5.14) de [Rucker, 2011] décomposent B_χ avec un terme supplémentaire relatif à une variable notée $g(s)$, équivalente à notre variable $T_0(s)$. Puisque le vecteur d'état $y(s)$ inclut systématiquement les variables de positions $p_0(s)$ et d'orientation $R_0(s)$, il y a donc une redondance entre les décompositions par rapport à $y(L_0)$ et $g(L_0)$. Finalement, le terme relatif à $g(L_0)$ ne devrait donc pas être présent dans l'expression de B_χ .
- La troisième différence apparaît dans le type de dérivées utilisées pour calculer B_χ . L'équation (2.65) montre que b inclut de manière explicite les efforts extérieurs $w_0(L_0)$ en bout : cela permet une dérivation directe pour le calcul de B_χ dans le cas de [Rucker and Webster, 2011]. Dans notre cas, nous considérons des efforts extérieurs $w_0(s_0 \neq L_0)$ qui n'apparaissent pas directement dans l'expression (2.65).

Si l'on applique naïvement la méthode de [Rucker and Webster, 2011] au cas de la compliance généralisée, il faudrait alors appliquer cette méthode autant de fois qu'il y a de points de discrétisation, soit N_P fois. Pour éviter d'alourdir inutilement ce calcul, nous réalisons les dérivations par rapport aux efforts extérieurs $w_0(s_0 \neq L_0)$ de manière indirecte au travers des dérivées $\frac{\partial y}{\partial \chi}(L_0)$.

Pour pouvoir utiliser l'équation (3.18b), il est encore nécessaire de calculer la variable $E_\chi(s)$. Or nous ne disposons pas d'expression directe pour la calculer. En utilisant la même approche que pour $V_\chi(s)$, il est possible de construire une équation différentielle pour estimer la propagation de la dérivée $E_\chi(s)$. En dérivant la définition (3.11) par rapport à s , en exploitant la linéarité de l'opérateur \bullet^\vee , en rappelant que s et χ sont indépendants puis en utilisant l'égalité (2.8b), on obtient le développement (3.25).

$$\dot{E}_\chi(s) = \frac{d}{ds} \left(\frac{\partial T_0^\vee(s)}{\partial \chi} \right) = \frac{d}{ds} \left(\frac{\partial T_0(s)}{\partial \chi} \right)^\vee = \frac{\partial}{\partial \chi} \left(\frac{dT_0(s)}{ds} \right)^\vee = \frac{\partial}{\partial \chi} \left(T_0(s) \hat{\xi}_0(s) \right)^\vee \quad (3.25)$$

La règle de dérivation d'une multiplication de fonctions nous permet de séparer la dernière expression de (3.25) pour obtenir (3.26)¹.

$$\dot{E}_\chi(s) = \left(\frac{\partial T_0}{\partial \chi}(s) \hat{\xi}_0(s) + T_0(s) \frac{\partial \hat{\xi}_0}{\partial \chi}(s) \right)^\vee \quad (3.26)$$

Finalement, on voit apparaître le terme $\hat{E}_\chi(s)$ dans (3.26) ce qui nous donne l'équation différentielle suivante :

$$\dot{E}_\chi(s) = \left(\hat{E}_\chi(s) \hat{\xi}_0(s) \right)^\vee + \left(T_0(s) \frac{\partial \hat{\xi}_0}{\partial \chi}(s) \right)^\vee \quad (3.27)$$

Les équations (3.22) et (3.27) permettent donc de calculer par propagation de dérivées les variables $V_\chi(s)$ et $E_\chi(s)$. La valeur $V_\chi(s = L_0)$ obtenue sera ensuite ré-utilisée pour calculer B_χ grâce à l'équation directe (3.24). Finalement, la compliance généralisée $C_{s_0}(s)$ pourra être calculée à partir de l'équation (3.18b), comme pour la MDFPVI. Cette méthode par propagation de dérivées est purement littérale et est donc exacte dans le sens où elle n'inclut pas d'erreur d'estimation comme pour les accroissements finis. De plus, elle réduit le nombre d'opérations nécessaires puisqu'elle n'intègre que les dérivées, et non l'ensemble du vecteur d'état $y(s)$. Afin éviter toute confusion avec la méthode suivante, les équations différentielles (3.22) et (3.27) seront désignées comme des équations "haut-niveau" et cette méthode sera désignée par l'acronyme MPDHN (Méthode par Propagation de Dérivées Haut-Niveau). Le nombre total d'opérations nécessaires pour estimer $C_{s_0}(s)$ par propagation de dérivées haut-niveau, noté N_{HN}^{PD} , peut être estimé par l'équation (3.28b).

1. Puisque χ est une variable muette permettant de désigner n'importe quel dérivateur, on considère ici que χ est un scalaire. Ainsi les matrices $\frac{\partial T_0}{\partial \chi}(s)$ et $\frac{\partial \hat{\xi}_0}{\partial \chi}(s)$ sont de taille 4×4 et l'opérateur \bullet^\vee peut être appliqué.

$$N_{HN}^{PD} = 6 \cdot N_P \cdot (Y N_P + B + 6N_P) + U \cdot (Y N_P + B + 6N_P) + 36N_P^2 \quad (3.28a)$$

$$= N_P^2 \cdot (6Y + 72) + N_P \cdot (6B + UY + 6U) + UB \quad (3.28b)$$

où \bullet^{PD} indique que le calcul est réalisé à partir d'une Propagation de Dérivées et \bullet_{HN} précise que cette propagation est réalisée sur les équations différentielles Haut-Niveau (3.22) et (3.27).

On observe que le terme dominant passe de $N_P^2 \cdot (12Y + 108)$ pour la méthode MDFPVI à $N_P^2 \cdot (6Y + 72)$ pour la méthode présentée ici. Cela signifie que, pour un même nombre de points de discrétisation N_P , la méthode de propagation des dérivées sera d'autant plus efficace que le vecteur d'état à intégrer sera de grande dimension. Bien que cette méthode de propagation de dérivées soit plus efficace et plus précise que les deux méthodes utilisant les accroissements finis, on peut se demander s'il ne serait pas possible de réduire d'avantage le nombre d'opérations nécessaires pour calculer $C_{s_0}(s)$. En effet, on rappelle que la propagation du vecteur d'état $y(s)$ peut être décrite par une unique équation différentielle (2.19) tandis que le calcul de la compliance nécessite, selon la MPDHN, l'intégration des deux équations différentielles (3.22) et (3.27). Il y a donc une redondance sur l'information intégrée : d'après la définition (3.20), $V_\chi(s)$ inclut la dérivée $\frac{\partial T_0^V}{\partial \chi}(s)$. Or cette dérivée est également incluse dans $E_\chi(s)$ d'après la définition (3.11). On peut alors constater que l'information relative à $\frac{\partial T_0^V}{\partial \chi}(s)$ est intégrée deux fois à travers les équations (3.22) et (3.27). En enlevant cette redondance, il serait a priori possible de réduire le nombre d'opérations et ainsi rendre le calcul de la jacobienne et de la matrice de compliance plus efficace.

3.2.4 Calcul littéral par propagation de dérivées bas-niveau

Nous proposons ici une nouvelle méthode de calcul par propagation de dérivées, désignée comme "bas-niveau" par opposition aux dérivées haut-niveau de la MPDHN. Puisque la forme du robot peut être décrite grâce à une unique équation différentielle (2.19), il paraît intéressant d'appliquer directement la dérivation par rapport à χ sur l'équation différentielle du vecteur d'état (2.19). Avec cette approche, l'information extraite par la dérivation est la plus compacte possible et permet alors d'éviter toute forme de redondance (comme c'est le cas avec la MPDHN). En dérivant (2.19) par rapport à χ et en rappelant que χ est indépendant de s , on obtient le développement suivant :

$$\frac{\partial \mathcal{F}}{\partial \chi}(y, s) = \frac{\partial \dot{y}}{\partial \chi}(s) = \frac{\partial}{\partial \chi} \left(\frac{dy}{ds} \right) (s) = \frac{d}{ds} \left(\frac{\partial y}{\partial \chi} \right) (s) \quad (3.29)$$

Ce développement simple permet de calculer la dérivée du vecteur d'état $\frac{\partial y}{\partial \chi}(s)$ par propagation grâce à une unique équation différentielle (3.29). Cette formulation a également l'avantage de découpler les intégrations des différentes composantes de $\frac{d}{ds} \left(\frac{\partial y}{\partial \chi} \right) (s)$. Comme présenté dans la sous-section 2.2.5, les composantes du vecteur d'état, dans le cas

des RTC, n'ont pas avoir les mêmes intervalles d'intégration. L'approche présentée permet donc d'éviter l'intégration de certaines composantes de $\frac{d}{ds}\left(\frac{\partial y}{\partial \chi}\right)(s)$ sur des intervalles qui ne les concernent pas.

L'idée d'économiser des opérations d'intégration inutiles a été approfondie et a permis une réduction importante du nombre total d'opérations. Lorsque l'on calcule la dérivée $\frac{d}{ds}\left(\frac{\partial y}{\partial w_0(s_0)}\right)(s)$, la dérivée $\frac{\partial w_0}{\partial w_0(s_0)}(s)$ intervient dans le calcul de $\frac{\partial \mathcal{F}}{\partial w_0(s_0)}(y, s)$ (équation (3.37) présentée plus tard pour les RTC). Il est clair que la dérivée $\frac{\partial w_0}{\partial w_0(s_0)}(s)$ est toujours nulle sauf lorsque $s = s_0$:

$$\forall (s, s_0) \in [0, L_0]^2, \frac{\partial w_0}{\partial w_0(s_0)}(s) = \begin{cases} 0_{6 \times 6} & \text{si } s \neq s_0 \\ I_{6 \times 6} & \text{si } s = s_0 \end{cases} \quad (3.30)$$

ce qui se répercute sur $\mathcal{F}(y, s)$ par :

$$\forall (s, s_0) \in [0, L_0]^2 \text{ tel que } s \neq s_0, \frac{\partial \mathcal{F}}{\partial w_0(s_0)}(y, s) = 0_{Y \times 6} \quad (3.31)$$

A cause de l'égalité (3.31), on sait que $\frac{\partial y}{\partial w_0(s_0)}(s)$ est constant sur le premier intervalle $[0, s_0]$ et ne peut évoluer que sur le second intervalle $[s_0, L_0]$. Si un effort $w_0(s_0 = 0)$ est appliqué à la base du robot alors c'est le mécanisme d'actionnement qui va absorber cet effort et non le robot qui est encore contraint en $s = 0$. Ainsi, on peut affirmer que la valeur initiale $\frac{\partial y}{\partial w_0(s_0)}(s = 0)$ est systématiquement nulle :

$$\forall s_0 \in [0, L_0], \frac{\partial y}{\partial w_0(s_0)}(0) = 0_{Y \times 6} \quad (3.32)$$

En exploitant les deux équations (3.31) et (3.32), on peut alors anticiper les valeurs de $\frac{\partial y}{\partial w_0(s_0)}(s)$ sur l'intervalle $[0, s_0]$:

$$\forall (s, s_0) \in ([0, s_0] \times [0, L_0]), \frac{\partial y}{\partial w_0(s_0)}(s) = \int_{h=0}^s \frac{\partial \dot{y}}{\partial w_0(s_0)}(h) dh + \frac{\partial y}{\partial w_0(s_0)}(0) \quad (3.33a)$$

$$= \int_{h=0}^s \frac{\partial \mathcal{F}}{\partial w_0(s_0)}(y, h) dh + \frac{\partial y}{\partial w_0(s_0)}(0) \quad (3.33b)$$

$$= 0_{Y \times 6} + 0_{Y \times 6} \quad (3.33c)$$

$$\forall (s, s_0) \in ([0, s_0] \times [0, L_0]), \frac{\partial y}{\partial w_0(s_0)}(s) = 0_{Y \times 6} \quad (3.33d)$$

où h est une variable muette utilisée à la place de s sous l'intégrale.

L'égalité (3.33) nous donne une information précieuse pour réduire les opérations de calcul : la propagation des dérivées associées à $\frac{\partial y}{\partial w_0(s_0)}(s)$ peut se restreindre à l'intervalle $[s_0, L_0]$ sans perdre d'information. Les intervalles d'intégration et de non-intégration des dérivées sont illustrés par la [figure 3.4](#). On remarquera que plus le point d'application

3.2. Calcul de la compliance généralisée

des efforts $w_0(s_0)$ est proche de l'extrémité $s = L_0$, plus l'intervalle d'intégration associé $[s_0, L_0]$ est petit.

L'économie sur le nombre d'opérations ne concerne que les dérivées par rapport à $w_0(s_0)$ et ne concernent pas celles par rapport à q et $y_u(0)$. Cependant, l'économie consistant à propager les dérivées grâce à une unique équation différentielle (et non grâce à deux équations comme pour la MPDHN) concerne elle toutes les dérivations.

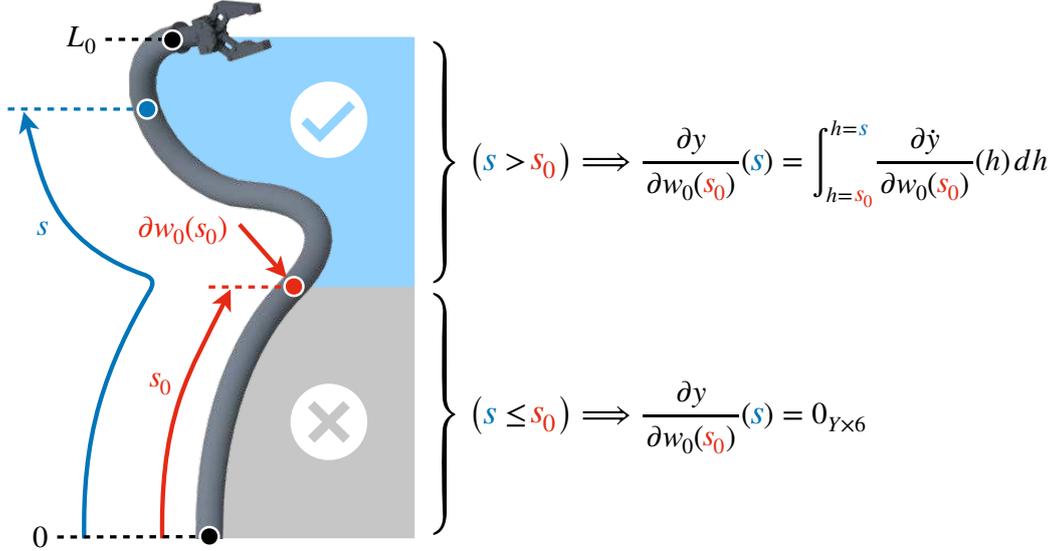


FIGURE 3.4 – Illustration de la réduction des intervalles d'intégration des dérivées partielles en fonction des points d'application des efforts externes.

Après avoir intégré l'équation (3.29), il est nécessaire de calculer $E_\chi(s)$ et $B_\chi(s)$ avant d'utiliser l'équation (3.18b) pour calculer la compliance $C_{s_0}(s)$. La variable $E_\chi(s)$ ne nécessite pas de calcul supplémentaire puisque les dérivées $\frac{\partial R_\chi^V}{\partial \chi}(s)$ et $\frac{\partial p_0}{\partial \chi}(s)$ sont déjà incluses dans $\frac{\partial y}{\partial \chi}(s)$. Le calcul de $B_\chi(s)$ nécessite quant à lui B opérations pour chaque dérivateur χ . Finalement, le nombre d'opérations nécessaire pour cette nouvelle méthode, désignée sous l'acronyme MPDBN (Méthode par Propagation de Dérivées Bas-Niveau), peut être estimé par l'équation (3.34b) :

$$N_{BN}^{PD} = 6 \cdot (Y(N_P + (N_P - 1) + \dots + 1) + BN_P) + U \cdot (YN_P + B) + 36N_P^2 \quad (3.34a)$$

$$= N_P^2 \cdot (3Y + 36) + N_P \cdot (6B + UY + 3Y) + UB \quad (3.34b)$$

où \bullet_{BN} précise que cette propagation est réalisée sur l'équation différentielle Bas-Niveau (3.29).

Dans le but de comparer notre nouvelle méthode à celles de la littérature, nous proposons d'estimer les gains d'opérations relatifs définis par (3.35) :

$$G_O(\text{MDFPVL}) = \frac{\|N_{PVL}^{DF} - N_{BN}^{PD}\|}{\|N_{PVL}^{DF}\|} \quad (3.35a)$$

$$G_O(\text{MDFPVI}) = \frac{\|N_{PVI}^{DF} - N_{BN}^{PD}\|}{\|N_{PVI}^{DF}\|} \quad (3.35b)$$

$$G_O(\text{MPDHN}) = \frac{\|N_{HN}^{PD} - N_{BN}^{PD}\|}{\|N_{HN}^{PD}\|} \quad (3.35c)$$

Ces gains ont été calculés pour plusieurs exemples de modèles et de robots présentés dans la littérature. Les paramètres des modèles et les résultats obtenus sont regroupés respectivement dans les tableaux [tableau 3.1](#) et [tableau 3.2](#). Ces résultats nous montrent que la MPDBN permet d'économiser, presque indépendamment des paramètres du modèle utilisé, jusqu'à 98% d'opérations par rapport à la MDFPVL, 73% par rapport à la MDFPVI et 49% par rapport à la MPDHN. Bien que les estimations du nombre d'opérations (3.7b), (3.19b), (3.28b) et (3.34b) ne prennent pas en compte certaines opérations comme celles liées à la manipulation de variables logicielles par exemple, on peut tout de même espérer obtenir des gains de temps de calcul similaires aux gains d'opérations. Une comparaison pratique des temps de calcul sera présentée dans le cas des RTC dans la [sous-section 3.3.4](#).

Référence	Type de robot	N_P	Y	U	B
Modèle du chapitre 2	RTC (2 tubes)	100	16	8	8
Modèle du chapitre 2	RTC (3 tubes)	100	18	9	9
[Rucker and Webster, 2011]	RTC (2 tubes)	60	22	6	5
[Rucker and Webster, 2011]	RTC (3 tubes)	60	24	9	6
[Tummers et al., 2023]	RC (6 câbles et 3 segments)	30	18	6	6
[Janabi-Sharifi et al., 2021]	RC (3 câbles et 1 segment)	40	27	6	6
[Childs and Rucker, 2023]*	RAA (2 segments)	100	40	10	4
[Childs and Rucker, 2023]*	RAA (3 segments)	100	53	12	8
[Chen et al., 2021b]*	RV (8 tiges et 2 segments)	100	26	10	10
[Chen et al., 2021b]*	RV (12 tiges et 3 segments)	100	30	12	12

TABLE 3.1 – Tableau regroupant des valeurs de N_P , Y , B et U pour différents modèles de robots continus présentés dans la littérature. Les références ne précisant pas le nombre de points de discrétisation sont indiquées par la mention •* et la valeur par défaut $N_P = 100$ est alors utilisée.

3.2. Calcul de la compliance généralisée

Référence	Gain d'opérations		
	G_O (MDFPVL)	G_O (MDFPVI)	G_O (MPDHN)
Modèle du chapitre 2	98%	72%	49%
Modèle du chapitre 2	98%	72%	49%
[Rucker and Webster, 2011]	98%	72%	49%
[Rucker and Webster, 2011]	98%	72%	49%
[Tummers et al., 2023]	98%	72%	49%
[Janabi-Sharifi et al., 2021]	98%	72%	49%
[Childs and Rucker, 2023]*	98%	73%	49%
[Childs and Rucker, 2023]*	98%	73%	49%
[Chen et al., 2021b]*	98%	73%	49%
[Chen et al., 2021b]*	98%	73%	49%

TABLE 3.2 – Tableau regroupant les estimations du nombre d'opérations pour les exemples référencés dans le [tableau 3.1](#) ainsi que les gains d'opérations permis par la méthode MPDBN proposée, relativement à la méthode MDFPVL, à la MDFPVI et à la MPDHN.

3.3 Cas d'étude : Modéliser et calculer la compliance généralisée des robots à tubes concentriques

Dans la section précédente, nous avons présenté comment la MPDBN permet de calculer efficacement la matrice de compliance généralisée. Dans cette section, nous allons détailler l'implémentation de cette méthode de calcul à la modélisation des RTC développée lors du [chapitre 2](#). Nous commencerons par présenter la propagation des dérivées bas-niveau dans la [sous-section 3.3.1](#) puis nous expliquerons comment les utiliser pour calculer les dérivées du modèle dans la [sous-section 3.3.2](#). Une validation numérique sera menée dans la [sous-section 3.3.3](#) pour estimer les performances de notre nouvelle méthode de calcul puis une comparaison des temps de calcul des différentes méthodes sera présentée dans la [sous-section 3.3.4](#).

3.3.1 Propagation des dérivées bas-niveau

Comme présenté dans la [sous-section 3.2.4](#), pour pouvoir appliquer la MPDBN à notre cas d'étude, il est nécessaire de dériver \mathcal{F} par rapport à χ . En appliquant les règles standards de dérivation sur l'équation (2.55) et (2.56), on obtient la forme développée de $\frac{\partial \mathcal{F}}{\partial \chi}(y, s)$ (3.37) ainsi que les dérivées des équations complémentaires (3.38).

Il convient de préciser ici une subtilité de dérivation concernant l'équation (3.38d). L'équation (2.56e) nous indique que $\phi_i(s)$ est discontinue. Or le point de discontinuité $s \in \{\max(0, \beta_{ci} - L_{ci}), \beta_{ci}\}$ dépend directement du dérivateur β_{ci} . En conséquence, la dérivée $\frac{\partial \phi_i}{\partial \chi}(s)$ n'est pas définie sur ce même point de discontinuité. Il reste toutefois possible d'estimer $\frac{\partial \phi_i}{\partial \chi}(s)$ à proximité de la discontinuité en utilisant localement une approximation par différences finies sur $\phi_i(s)$ puisqu'en pratique l'abscisse curviligne s est discrétisée en s_i où $i \in \llbracket 1, N_P \rrbracket$. De plus, puisque $\phi_i(s)$ intervient dans les expressions de $u_i^*(s)$, $u_i(s)$ et $u_0(s)$, on pourra appliquer la même approche pour ces autres dérivées. On comprend alors que la valeur du pas de discrétisation, noté $\Delta(s)$, influence directement la valeur de cette dérivée : plus la discrétisation $\Delta(s)$ sera fine, plus la valeur de $\frac{\partial \phi_i}{\partial \chi}(s)$ sera grande. Pour une application pratique, lorsque la jacobienne $J(s)$ est utilisée pour commander les translations des tubes β_{ci} , il est conseillé de saturer $\|\Delta \beta_{ci}\| \leq \max(\Delta(s))$ afin d'éviter l'envoi de consignes aberrantes aux actionneurs du robot.

On pourra également remarquer la présence des dérivées $\frac{\partial \dot{\tau}_0}{\partial \chi}(s)$ et $\frac{\partial \dot{f}_0}{\partial \chi}(s)$ dans les équations (3.37c) et (3.37d). Les termes $\dot{\tau}_0(s)$ et $\dot{f}_0(s)$ représentent les distributions linéaires respectivement des couples $\tau_0(s)$ et des forces $f_0(s)$ appliqués sur le robot. Lorsque l'on considère un effort externe ponctuel, comme c'est le cas lorsque l'on dérive les équations (2.55c) et (2.55d) par rapport à $w_0(s_0)$, on peut alors reprendre la description utilisée dans la [sous-section 2.1.4](#) et définir ces distributions comme les ratios entre les efforts et la longueur du pas de discrétisation :

$$\dot{\tau}_0(s) = \frac{\tau_0(s)}{\Delta(s)} \text{ et } \dot{f}_0(s) = \frac{f_0(s)}{\Delta(s)} \quad (3.36)$$

$$\frac{d}{ds} \left(\frac{\partial \theta_i}{\partial \chi} \right) (s) = \frac{\partial u_i|_z}{\partial \chi} (s) \quad (3.37a)$$

$$\begin{aligned} \frac{d}{ds} \left(\frac{\partial u_i|_z}{\partial \chi} \right) (s) &= \frac{kx_i}{kz_i} \left(\frac{\partial u_0}{\partial \chi} (s)^T \text{Rot}_z(\theta_i(s)) \begin{bmatrix} u_i^*(s)|_y \\ -u_i^*(s)|_x \\ 0 \end{bmatrix} \right. \\ &\quad \left. + u_0(s)^T \hat{e}_z \text{Rot}_z(\theta_i(s)) \frac{\partial \theta_i}{\partial \chi} (s) \begin{bmatrix} u_i^*(s)|_y \\ -u_i^*(s)|_x \\ 0 \end{bmatrix} \right. \\ &\quad \left. + u_0(s)^T \text{Rot}_z(\theta_i(s)) \begin{bmatrix} \frac{\partial u_i^*|_y}{\partial \chi} (s) \\ -\frac{\partial u_i^*|_x}{\partial \chi} (s) \\ 0 \end{bmatrix} \right) \end{aligned} \quad (3.37b)$$

$$\begin{aligned} \frac{d}{ds} \left(\frac{\partial m_0}{\partial \chi} \right) (s) &= \frac{\partial \hat{u}_0}{\partial \chi} (s) m_0(s) + \hat{u}_0(s) \frac{\partial m_0}{\partial \chi} (s) - \hat{e}_z \frac{\partial n_0}{\partial \chi} (s) - \frac{\partial R_0}{\partial \chi} (s)^T \dot{\tau}_0(s) \\ &\quad - R_0(s)^T \frac{\partial \dot{\tau}_0}{\partial \chi} (s) \end{aligned} \quad (3.37c)$$

$$\frac{d}{ds} \left(\frac{\partial n_0}{\partial \chi} \right) (s) = \frac{\partial \hat{u}_0}{\partial \chi} (s) n_0(s) + \hat{u}_0(s) \frac{\partial n_0}{\partial \chi} (s) - \frac{\partial R_0}{\partial \chi} (s)^T \dot{f}_0(s) - R_0(s)^T \frac{\partial \dot{f}_0}{\partial \chi} (s) \quad (3.37d)$$

$$\frac{d}{ds} \left(\frac{\partial R_0}{\partial \chi} \right) (s) = \frac{\partial R_0}{\partial \chi} (s) \hat{u}_0(s) + R_0(s) \frac{\partial \hat{u}_0}{\partial \chi} (s) \quad (3.37e)$$

$$\frac{d}{ds} \left(\frac{\partial p_0}{\partial \chi} \right) (s) = \frac{\partial R_0}{\partial \chi} (s) e_z \quad (3.37f)$$

$$\begin{aligned} \frac{\partial u_0}{\partial \chi} (s) \Big|_{x,y} &= \left(\frac{\partial}{\partial \chi} \left(\sum_{i \in nbT(s)} K_i \right)^{-1} \sum_{i \in nbT(s)} \left(\text{Rot}_z(\theta_i(s)) K_i u_i^*(s) + m_0(s) \right) \right. \\ &\quad \left. + \left(\sum_{i \in nbT(s)} K_i \right)^{-1} \sum_{i \in nbT(s)} \left(\hat{e}_z \text{Rot}_z(\theta_i(s)) \frac{\partial \theta_i}{\partial \chi} (s) K_i u_i^*(s) \right. \right. \\ &\quad \left. \left. + \text{Rot}_z(\theta_i(s)) K_i \frac{\partial u_i^*}{\partial \chi} (s) + \frac{\partial m_0}{\partial \chi} (s) \right) \right) \Big|_{x,y} \end{aligned} \quad (3.38a)$$

$$\frac{\partial u_i}{\partial \chi} (s) \Big|_{x,y} = \left(\left(\hat{e}_z \text{Rot}_z(\theta_i(s)) \frac{\partial \theta_i}{\partial \chi} (s) \right)^T u_0(s) + \text{Rot}_z(\theta_i(s)) \frac{\partial u_0}{\partial \chi} (s) \right) \Big|_{x,y} \quad (3.38b)$$

$$\frac{\partial u_i^*}{\partial \chi} (s) = \hat{e}_z \text{Rot}_z(\theta_{ci}) \frac{\partial \theta_{ci}}{\partial \chi} \begin{bmatrix} \phi_i(s) \\ 0 \\ 0 \end{bmatrix} + \text{Rot}_z(\theta_{ci}) \begin{bmatrix} \frac{\partial \phi_i}{\partial \chi} (s) \\ 0 \\ 0 \end{bmatrix} \quad (3.38c)$$

$$\frac{\partial \phi_i}{\partial \chi} (s) = \begin{cases} \pm \infty & \text{si } \chi = \beta_{ci} \text{ et } s \in \{\max(0, \beta_{ci} - L_{ci}), \beta_{ci}\} \\ 0 & \text{sinon} \end{cases} \quad (3.38d)$$

A partir des équations (3.36), il est alors possible de calculer les termes $\frac{\partial \tau_0}{\partial \chi}(s)$ et $\frac{\partial f_0}{\partial \chi}(s)$. L'ensemble des valeurs particulières des dérivées ainsi que les valeurs initiales non nulles impliquées dans les équations (3.37a) à (3.38d) ont été regroupées dans les tableaux 3.3 et 3.4.

Valeurs initiales		
Dérivée	Abscisse s	Valeur
$\frac{\partial u_i(s) _z}{\partial u_j(0_j) _z}$	$s = 0_i$	δ_{ij}
$\frac{\partial \theta_{ci}}{\partial \theta_{cj}}$	$s = 0_i$	δ_{ij}
$\frac{\partial m_0}{\partial m_0(0)}(s)$	$s = 0$	I_3
$\frac{\partial n_0}{\partial n_0(0)}(s)$	$s = 0$	I_3

TABLE 3.3 – Synthèse des valeurs initiales des dérivées impliquées dans le calcul de $\frac{\partial y}{\partial \chi}(s)$.

Valeurs particulières		
Dérivée	Abscisse s	Valeur
$\frac{\partial \phi_i}{\partial \beta_{cj}}(s)$	$s = \beta_{ci} - L_{ci}$ (si $\beta_{ci} - L_{ci} \geq 0$)	$-\delta_{ij} \frac{R_{ci}}{\Delta(s)}$
$\frac{\partial u_i}{\partial \beta_{cj}}(s) _{x,y}$	$s = \beta_{ci} - L_{ci}$ (si $\beta_{ci} - L_{ci} \geq 0$)	$-\delta_{ij} \frac{u_i(s+\Delta(s)) - u_i(s-\Delta(s))}{2 \cdot \Delta(s)} _{x,y}$
$\frac{\partial u_i^*}{\partial \beta_{cj}}(s)$	$s = \beta_{ci} - L_{ci}$ (si $\beta_{ci} - L_{ci} \geq 0$)	$-\delta_{ij} \frac{u_i^*(s+\Delta(s)) - u_i^*(s-\Delta(s))}{2 \cdot \Delta(s)}$
$\frac{\partial u_0}{\partial \beta_{cj}}(s) _{x,y}$	$s = \beta_{ci} - L_{ci}$ (si $\beta_{ci} - L_{ci} \geq 0$)	$-\frac{u_0(s+\Delta(s)) - u_0(s-\Delta(s))}{2 \cdot \Delta(s)} _{x,y}$
$\frac{\partial \phi_i}{\partial \beta_{cj}}(s)$	$s = \beta_{ci}$	$\delta_{ij} \frac{R_{ci}}{\Delta(s)}$
$\frac{\partial u_i}{\partial \beta_{cj}}(s) _{x,y}$	$s = \beta_{ci}$	$\delta_{ij} \frac{u_i(s+\Delta(s)) - u_i(s-\Delta(s))}{2 \cdot \Delta(s)} _{x,y}$
$\frac{\partial u_i^*}{\partial \beta_{cj}}(s)$	$s = \beta_{ci}$	$\delta_{ij} \frac{u_i^*(s+\Delta(s)) - u_i^*(s-\Delta(s))}{2 \cdot \Delta(s)}$
$\frac{\partial u_0}{\partial \beta_{cj}}(s) _{x,y}$	$s = \beta_{ci}$	$\frac{u_0(s+\Delta(s)) - u_0(s-\Delta(s))}{2 \cdot \Delta(s)} _{x,y}$
$\frac{\partial \tau_0}{\partial \tau_0(s_0)}(s)$	$s = s_0$	I_3
$\frac{\partial f_0}{\partial f_0(s_0)}(s)$	$s = s_0$	I_3
$\frac{\partial \dot{\tau}_0}{\partial \tau_0(s_0)}(s)$	$s = s_0$	$\frac{I_3}{\Delta(s)}$
$\frac{\partial \dot{f}_0}{\partial f_0(s_0)}(s)$	$s = s_0$	$\frac{I_3}{\Delta(s)}$

TABLE 3.4 – Synthèse des valeurs particulières des dérivées impliquées dans le calcul de $\frac{\partial y}{\partial \chi}(s)$.

Certaines des valeurs initiales et particulières indiquées dans les tableaux 3.3 et 3.4 utilisent le delta de Kronecker, noté δ_{ij} , défini par :

$$\forall (i, j) \in \mathbb{N}^2, \delta_{ij} = \begin{cases} 1 & \text{si } i = j \\ 0 & \text{sinon} \end{cases} \quad (3.39)$$

3.3.2 Calcul des dérivées haut-niveau

Comme présenté dans la sous-section 3.2.4, l'intégration des dérivées partielles bas-niveau (3.37) permet d'extraire la totalité de l'information nécessaire pour calculer la compliance généralisée. Le lien entre les dérivées bas-niveau (issues de l'intégration) et la compliance $C_{s_0}(s)$ peut ensuite se faire grâce à l'équation haut-niveau (3.18b). Pour cela il est nécessaire de reconstruire les matrices haut-niveau $E_\chi(s)$ et B_χ à partir des valeurs de $\frac{\partial y}{\partial \chi}(s)$. La définition (3.11) montre que $E_\chi(s)$ ne nécessite pas d'opération spécifique puisque $E_\chi(s)$ est simplement la concaténation verticale des dérivées $\frac{\partial p_0}{\partial \chi}(s)$ et $\frac{\partial R_0^\vee}{\partial \chi}(s)$ incluses dans $\frac{\partial y}{\partial \chi}(s)$:

$$E_\chi(s) = \begin{bmatrix} \frac{\partial p_0}{\partial \chi}(s) \\ \frac{\partial R_0^\vee}{\partial \chi}(s) \end{bmatrix} \quad (3.40)$$

Pour le calcul de B_χ , il est toutefois nécessaire de développer l'expression dérivée du résidu b . A partir des règles usuelles de dérivation, la définition (3.13) se dérive par rapport à χ de la manière suivante :

$$B_\chi = \begin{bmatrix} \frac{\partial u_1}{\partial \chi}(\beta_{c1})|_z \\ \dots \\ \frac{\partial u_i}{\partial \chi}(\beta_{ci})|_z \\ \dots \\ \frac{\partial u_{nbT-1}}{\partial \chi}(\beta_{cnbT-1})|_z \\ \frac{\partial u_{nbT}}{\partial \chi}(L_0) - \frac{(0 \ 0 \ 1]}{kz_{nbT}} \left[\frac{\partial R_{nbT}}{\partial \chi}(L_0)^T \tau_0(L_0) + R_{nbT}(L_0)^T \frac{\partial \tau_0}{\partial \chi}(L_0) \right] \\ \frac{\partial m_0}{\partial \chi}(L_0) - \left(\frac{\partial R_0}{\partial \chi}(s)^T \tau_0(L_0) + R_0(L_0)^T \frac{\partial \tau_0}{\partial \chi}(L_0) \right) \\ \frac{\partial n_0}{\partial \chi}(L_0) - \left(\frac{\partial R_0}{\partial \chi}(s)^T f_0(L_0) + R_0(L_0)^T \frac{\partial f_0}{\partial \chi}(L_0) \right) \end{bmatrix} \quad (3.41)$$

où :

$$\frac{\partial R_{nbT}}{\partial \chi}(s) = \frac{\partial R_0}{\partial \chi}(s) Rot_z(\theta_{nbT}(s))^T + R_0(s) \left(\hat{e}_z Rot_z(\theta_{nbT}(s)) \frac{\partial \theta_{nbT}}{\partial \chi}(s) \right)^T \quad (3.42)$$

On remarquera qu'il est donc possible de reconstruire les dérivées haut-niveau $E_\chi(s)$ et B_χ à partir de la dérivée bas-niveau du vecteur d'état. Une fois ces calculs réalisés, il est alors possible d'utiliser la décomposition (3.18b) pour estimer complètement la compliance

$C_{s_0}(s)$. La jacobienne $J(s)$ nécessitera quant à elle l'utilisation de l'équation (3.18a). La MPDBN pour le calcul de la matrice de compliance généralisée $C_{s_0}(s)$ et de la jacobienne articulaire $J(s)$ est synthétisée en annexe par la figure 5.2.

3.3.3 Validation numérique

La MPDBN ainsi que les autres méthodes de calcul présentées dans la section 3.2 ont été implémentées sous Matlab. Dans cette section, nous allons présenter une comparaison numérique réalisée sur le RTC2 dans le but de quantifier les performances de la MPDBN. Cette vérification nécessite donc l'utilisation d'une autre méthode de calcul qui sera considérée comme une référence. Pour éviter d'introduire toute erreur d'implémentation dans la méthode de référence, nous avons utilisé la méthode la plus simple d'implémentation : la MDFPVL présentée dans la sous-section 3.2.1. Dans un premier temps, une comparaison des coefficients de $C_{s_0}(s)$ et $J(s)$ a été réalisée pour confronter les résultats de la MDFPVL et de la MPDBN. Cette approche a permis de visualiser graphiquement les valeurs des coefficients de ces deux matrices pour tous les $s \in [0, L_0]$ et $s_0 \in [0, L_0]$. Un exemple est présenté en annexe par la figure 5.3.

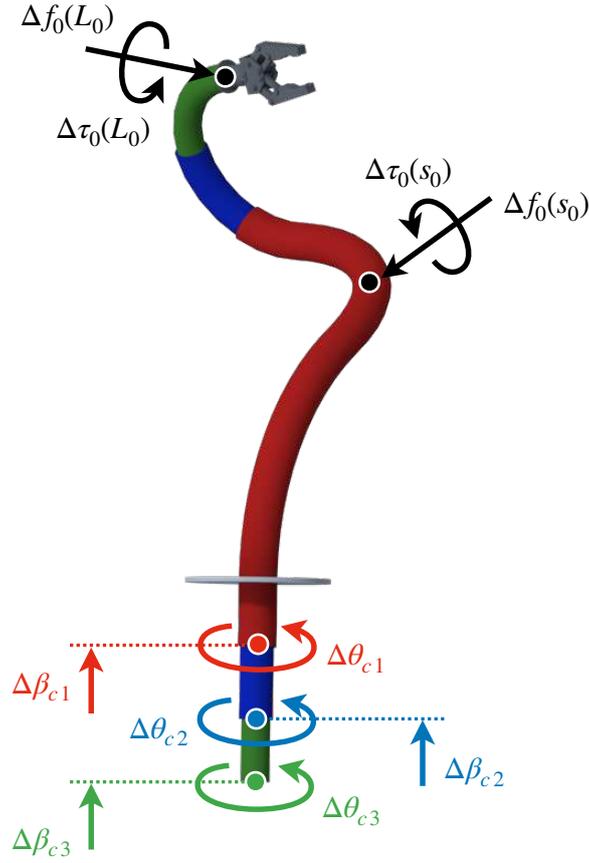
Ce type de comparaison a été utilisée pour détecter les erreurs lors de l'implémentation de la MPDBN mais est toutefois inadaptée pour une validation quantitative. En effet, les coefficients matriciels ont des unités diverses et donc des ordres de grandeurs parfois très différents. Pour cette raison, la comparaison matricielle est restée un outil de développement logiciel et c'est une comparaison en déformation du robot qui a été utilisée pour valider la MPDBN. Cette comparaison consiste à calculer de deux manières différentes une même déformation du robot, à partir d'une même forme initiale, notée $p^*(s)$ et calculée grâce à au modèle \mathcal{M} :

$$p^*(s) = \mathcal{M}(s, \Omega_{rob}, \Omega_{ctrl}) \quad (3.43)$$

où l'indice \bullet_0 est omis sur $p^*(s)$ pour simplifier les notations.

Connaissant $p^*(s)$ pour une valeur $(\Omega_{rob}, \Omega_{ctrl})$ donné, nous nous intéressons ensuite à quantifier la déformation du robot lorsqu'il est soumis à des variations d'actionnement Δq , d'effort sur le corps $\Delta w_0(s_0)$ et d'effort à l'extrémité $\Delta w_0(L_0)$. Dans cette étude, les efforts appliqués en $s_0 = L_0$ et $s_0 \neq L_0$ sont différenciés et peuvent donc être appliqués simultanément. L'ensemble de ces variations, illustrées par la figure 3.5, peuvent être regroupées sous une même variation noté $\Delta\Omega_{ctrl}$:

$$\Delta\Omega_{ctrl} = \{\Delta\theta_{ci}, \Delta\beta_{ci}, \Delta\tau_0(s_0), \Delta f_0(s_0), \Delta\tau_0(L_0), \Delta f_0(L_0)\} \quad (3.44)$$


 FIGURE 3.5 – Représentation des différentes variations regroupées dans $\Delta\Omega_{ctrl}$.

Pour quantifier la déformation du robot générée par la variation $\Delta\Omega_{ctrl}$, deux méthodes sont utilisées :

- La première méthode consiste à résoudre le nouveau PVL qui inclut la variation $\Delta\Omega_{ctrl}$. La nouvelle forme sera notée $p_{mod}(s)$:

$$p_{mod}(s) = \mathcal{M}(s, \Omega_{rob}, \Omega_{ctrl} + \Delta\Omega_{ctrl}) \quad (3.45)$$

- La seconde méthode consiste à utiliser les valeurs de $C_{s_0}(s)$ et $J(s)$ obtenues à partir de la MPDBN pour linéariser la déformation du robot qui sera alors ajoutée à la forme initiale $p^*(s)$. La nouvelle forme obtenue, notée $p_{jacob}(s)$, est définie par l'équation (3.46) où l'on précise avec $\bullet|_{p_0(s)}$ que l'on exploite que les valeurs des matrices correspondantes aux variations de position (et non d'orientation).

$$p_{jacob}(s) \approx p^*(s) + \sum_{s_0 \in [0, L_0]} C_{s_0}(s)|_{p_0(s)} \begin{bmatrix} \Delta\tau_0(s_0) \\ \Delta f_0(s_0) \end{bmatrix} + \sum_{i \in [1, nbT]} J(s)|_{p_0(s)} \begin{bmatrix} \Delta\theta_{ci} \\ \Delta\beta_{ci} \end{bmatrix} \quad (3.46)$$

Même si les composantes des matrices relatives aux dérivées d'orientation $\frac{\partial R_0^Y}{\partial \chi}(s)$ ne sont pas directement utilisées, elles sont tout de même couplées aux dérivées de position $\frac{\partial p_0}{\partial \chi}(s)$ (équation (3.37f)) et impliquées dans le calcul de B_χ (équation (3.41)). La validation proposée consiste alors à vérifier le réalisme du modèle de compliance en quantifiant le

réalisme des déformées linéarisées. Avec cette approche, nous utilisons une même mesure (une distance entre deux formes de robot) au lieu d'une mesure par coefficient de matrice. La forme $p_{mod}(s)$ issue de la résolution du modèle (3.45) sera prise en référence tandis que la forme $p_{jacob}(s)$ servira à estimer les performances de la MPDBN appliquée à $J(s)$ et $C_{s_0}(s)$. On propose alors de mesurer les distances respectives entre les trois formes $p^*(s)$, $p_{mod}(s)$ et $p_{jacob}(s)$. Puisque l'intensité d'une déformation dépend directement du point s considéré, on ne s'intéresse ici qu'à la distance maximale entre deux formes. En considérant deux formes $p_1(s)$ et $p_2(s)$, la mesure utilisée sera notée $d(p_1(s), p_2(s))$ et sera définie de la manière suivante :

$$d(p_1(s), p_2(s)) = \max_{s \in [0, L_0]} \|p_1(s) - p_2(s)\| \quad (3.47)$$

En pratique, la valeur maximale de $\|p_1(s) - p_2(s)\|$ est le plus souvent obtenue à l'extrémité $s = L_0$ du robot. Lorsqu'une variation $\Delta\beta_{ci}$ est appliquée, les deux formes $p_1(s)$ et $p_2(s)$ n'ont alors plus la même longueur. Si l'on compare directement les deux formes de longueur différentes, alors la mesure $d(p_1(s), p_2(s))$ risquerait d'être biaisée par une erreur minimale systématique associée au décalage de longueur. Pour éviter tout problème lié à l'application de la variation $\Delta\beta_{ci}$, nous avons alors choisi de restreindre la mesure $d(p_1(s), p_2(s))$ sur la longueur commune entre $p_1(s)$ et $p_2(s)$. Finalement, les trois distances différentes indiquées par (3.48) et illustrées par la figure 3.6 ont été mesurées pour notre étude. Sachant que l'amplitude des déformations croît avec la longueur du robot L_0 , ces mesures sont systématiquement normalisées par rapport à la longueur L_0 (ou par la longueur la plus restrictive le cas échéant).

$$\mathcal{D}_{mod}^* [\%] = \frac{d(p^*(s), p_{mod}(s))}{L_0} \cdot 100 \quad (3.48a)$$

$$\mathcal{D}_{jacob}^* [\%] = \frac{d(p^*(s), p_{jacob}(s))}{L_0} \cdot 100 \quad (3.48b)$$

$$\mathcal{D}_{mod}^{jacob} [\%] = \frac{d(p_{jacob}(s), p_{mod}(s))}{L_0} \cdot 100 \quad (3.48c)$$

L'objectif de la validation est d'utiliser des configurations aléatoires du robot, c'est-à-dire des valeurs aléatoires pour $\Delta\Omega_{ctrl}$, puis de comparer les distances relatives (3.48a)-(3.48c). Bien que la variation $\Delta\Omega_{ctrl}$ soit aléatoire, son intensité est limitée afin d'éviter des valeurs aberrantes (des forces trop intenses par rapport à la rigidité du robot par exemple). De plus, si l'une des variables de $\Delta\Omega_{ctrl}$ génère des déformations systématiquement plus grandes que les autres, l'évaluation de la validité du calcul risque d'être biaisée. Une étude préliminaire a permis de trouver des valeurs limites pour $\|\Delta\Omega_{ctrl}\|$ satisfaisantes, c'est-à-dire générant individuellement des déformations de même amplitude, comme illustré en annexe par la figure 5.4. Les valeurs limites $\|\Delta\Omega_{ctrl}\|$ retenues sont regroupées dans le tableau 3.5. On précisera que les efforts ponctuels sur le corps du robot sont appliqués sur le point d'abscisse curviligne $s_0 \in [0, L_0[$, évitant donc d'appliquer simultanément deux efforts au bout du robot.

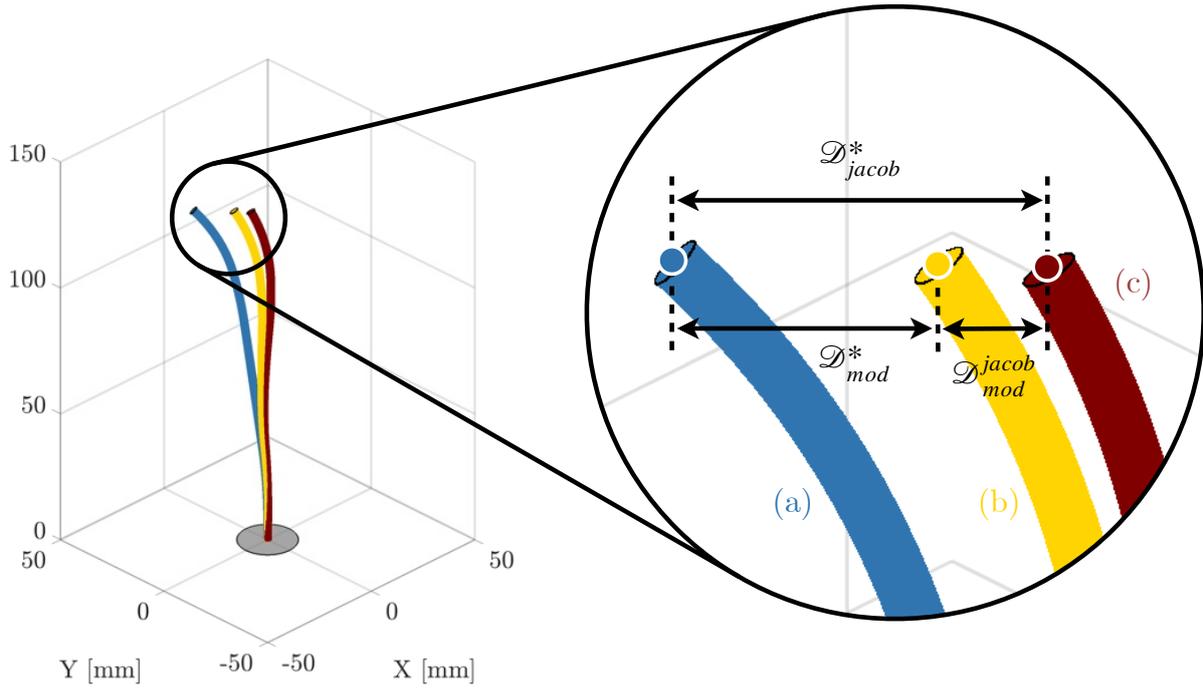


FIGURE 3.6 – Illustration des distances définies par (3.48) et mesurées entre la forme initiale $p^*(s)$ (a), la forme déformée calculée grâce au modèle cinétostatique $p_{mod}(s)$ (b) et la forme déformée calculée par linéarisation grâce à la matrice de compliance généralisée et la jacobienne $p_{jacob}(s)$ (c).

Variations limites			
Description	Variable	Unité	Maximum
Rotation des tubes	$\ \Delta\theta_{c_i}\ $	rad	$2 \cdot 10^{-6}$
Translation des tubes	$\ \Delta\beta_{c_i}\ $	mm	$5 \cdot 10^{-5}$
Couple à l'extrémité du robot	$\ \Delta\tau_0(L_0)\ $	N.mm	$8\sqrt{3} \cdot 10^{-1}$
Force à l'extrémité du robot	$\ \Delta f_0(L_0)\ $	N	$2\sqrt{3} \cdot 10^{-2}$
Couple ponctuel le long du robot	$\ \Delta\tau_0(s_0)\ $	N.mm	$4\sqrt{3} \cdot 10^{-1}$
Force ponctuelle le long du robot	$\ \Delta f_0(s_0)\ $	N	$8\sqrt{3} \cdot 10^{-3}$

TABLE 3.5 – Synthèse des variations maximales appliquées sur la forme initiale $p^*(s)$ lors de la validation, pour la configuration RTC2-C1. Ces variations sont à l'origine des formes déformées $p_{mod}(s)$ et $p_{jacob}(s)$.

Les variations aléatoires $\Delta\Omega_{ctrl}$ ont été ensuite appliquées au RTC présenté dans la sous-section 2.2.6 et pour différentes formes. Dans le but de quantifier les performances générales de la MPDBN, il convient de la tester sur une grande variété de formes. Ces dernières changent en fonction de l'actionnement et des efforts externes, regroupés dans

Ω_{ctrl} . Il est important de préciser que la valeur Ω_{ctrl} , qui définit la forme initiale $p^*(s)$, est indépendante de la variation $\Delta\Omega_{ctrl}$ qui définit quant à elle les formes déformées $p_{mod}(s)$ et $p_{jacob}(s)$. On distingue alors deux types de scénarios :

- Le cas "sans précharge" : les formes considérées ne sont pas soumises à des efforts externes et ne sont donc pas déformées avant l'application de variation $\Delta\Omega_{ctrl}$:

$$w_0(s_0) = 0_{6 \times 1} \text{ et } w_0(L_0) = 0_{6 \times 1} \quad (3.49)$$

- Le cas "avec précharge" : les formes considérées sont déjà déformées à cause d'un effort externe aléatoire réparti uniformément sur l'intervalle $s \in [l_{min}, l_{max}]$, avant l'application de variation $\Delta\Omega_{ctrl}$:

$$w_0(s_0) \neq 0_{6 \times 1} \text{ et } w_0(L_0) \neq 0_{6 \times 1} \quad (3.50)$$

On rappelle que la MPDBN tient compte d'éventuelles précharges lors du calcul de $C_{s_0}(s)$ et de $J(s)$. Comme pour les variations $\Delta\Omega_{ctrl}$, les valeurs de Ω_{ctrl} seront choisies aléatoirement pour générer une grande diversité de formes initiales $p^*(s)$. Les valeurs limites pour les variables d'actionnement θ_{ci} et β_{ci} ont été choisies de manière à couvrir la totalité de l'espace de travail. Les valeurs limites des précharges $w_0(s_0)$ et $w_0(L_0)$ ont été sélectionnées de manière à tester des formes largement déformées (par rapport à la longueur du robot) tout en garantissant que les intensités des efforts restent réalistes (pour un robot médical, ces efforts sont générés lors du contact avec des tissus biologiques mous). Dans ce but, une seconde étude préliminaire a permis de sélectionner des valeurs limites pour Ω_{ctrl} regroupées dans le tableau [tableau 3.6](#).

Couple et force limites de précharge			
Description	Variable	Unité	Maximum
Couple à l'extrémité du robot	$\ \tau_0(L_0)\ $	N.mm	$1 \cdot 10^2 \cdot \sqrt{3}$
Force à l'extrémité du robot	$\ f_0(L_0)\ $	N	$1 \cdot 10^0 \cdot \sqrt{3}$
Couple distribué le long du robot	$\ \tau_0(s_0)\ $	N.mm	$1 \cdot 10^3 \cdot \sqrt{3}$
Force distribuée le long du robot	$\ f_0(s_0)\ $	N	$1 \cdot 10^1 \cdot \sqrt{3}$
Longueur du segment d'application	$ l_{max} - l_{min} $	mm	$2 \cdot 10^1$

TABLE 3.6 – Synthèse des précharges limites appliquées permettant d'obtenir des formes initiales $p^*(s)$ déformées pour les scénarios avec précharge.

Pour les deux cas avec et sans précharge, 5000 formes $p^*(s)$ aléatoires ont été générées grâce à Ω_{ctrl} . Un extrait de 500 formes $p^*(s)$ obtenues dans le cas sans précharge est présenté par la [figure 3.7](#). Dans les deux cas, une variation aléatoire $\Delta\Omega_{ctrl}$ a été appliquée pour chaque forme $p^*(s)$ et les formes déformées obtenues $p_{mod}(s)$ et $p_{jacob}(s)$ ont permis de mesurer les distances \mathcal{D}_{mod}^* , \mathcal{D}_{jacob}^* et $\mathcal{D}_{mod}^{jacob}$. Les résultats obtenus sont présentés par la [figure 3.8](#) et les mesures statistiques extraites ont été synthétisées dans le [tableau 3.7](#).

3.3. Cas d'étude : Modéliser et calculer la compliance généralisée des RTC

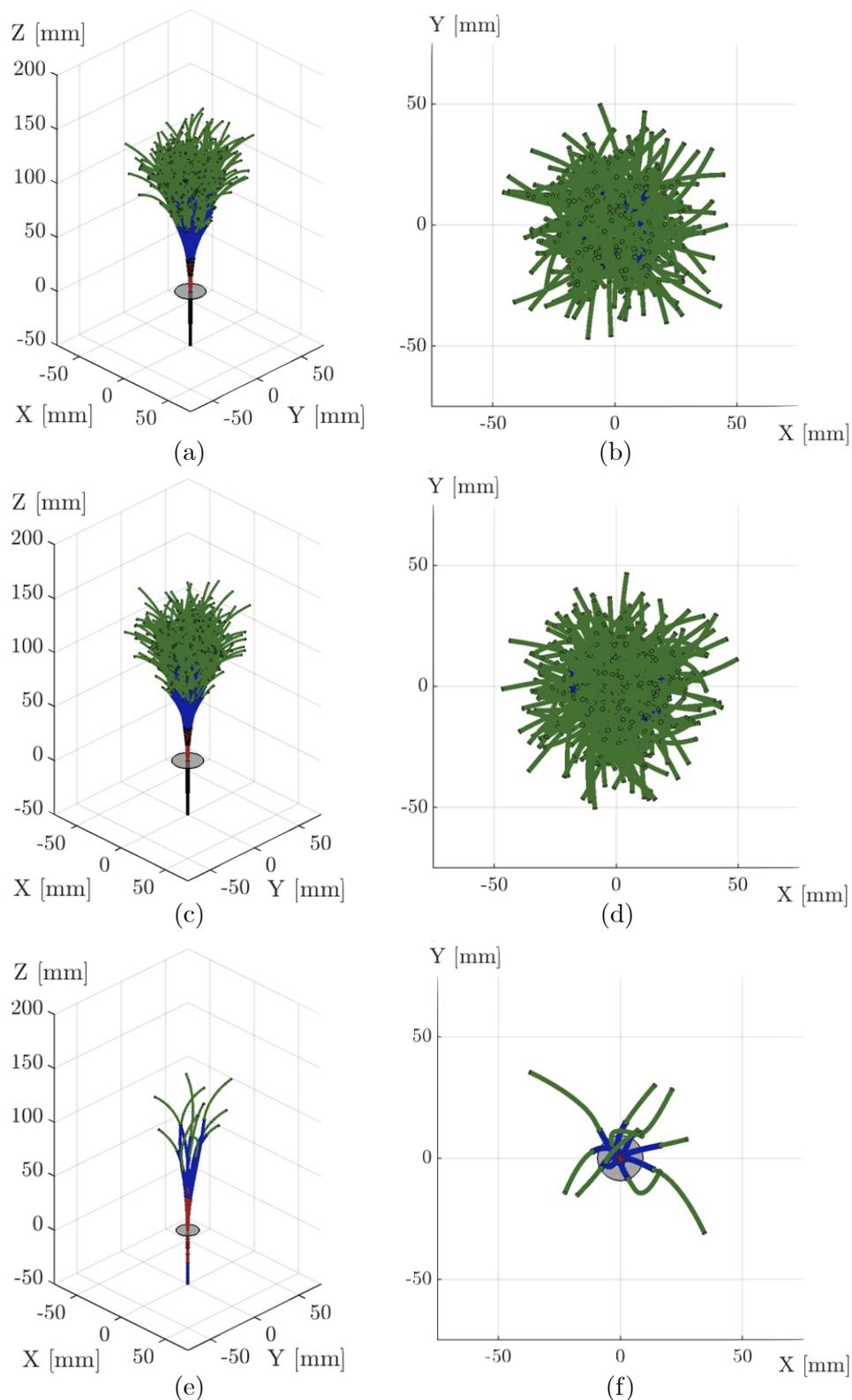


FIGURE 3.7 – Présentation d'un extrait de 500 formes $p^*(s)$ (sur un total de 5000 formes) simulées pour le cas sans précharge (a)-(b) et dans le cas avec précharge (c)-(d). L'extrait de 10 formes $p^*(s)$ pour le cas avec précharge (e)-(f) souligne la grande diversité des formes simulées incluant notamment des formes en "C" et en "S".

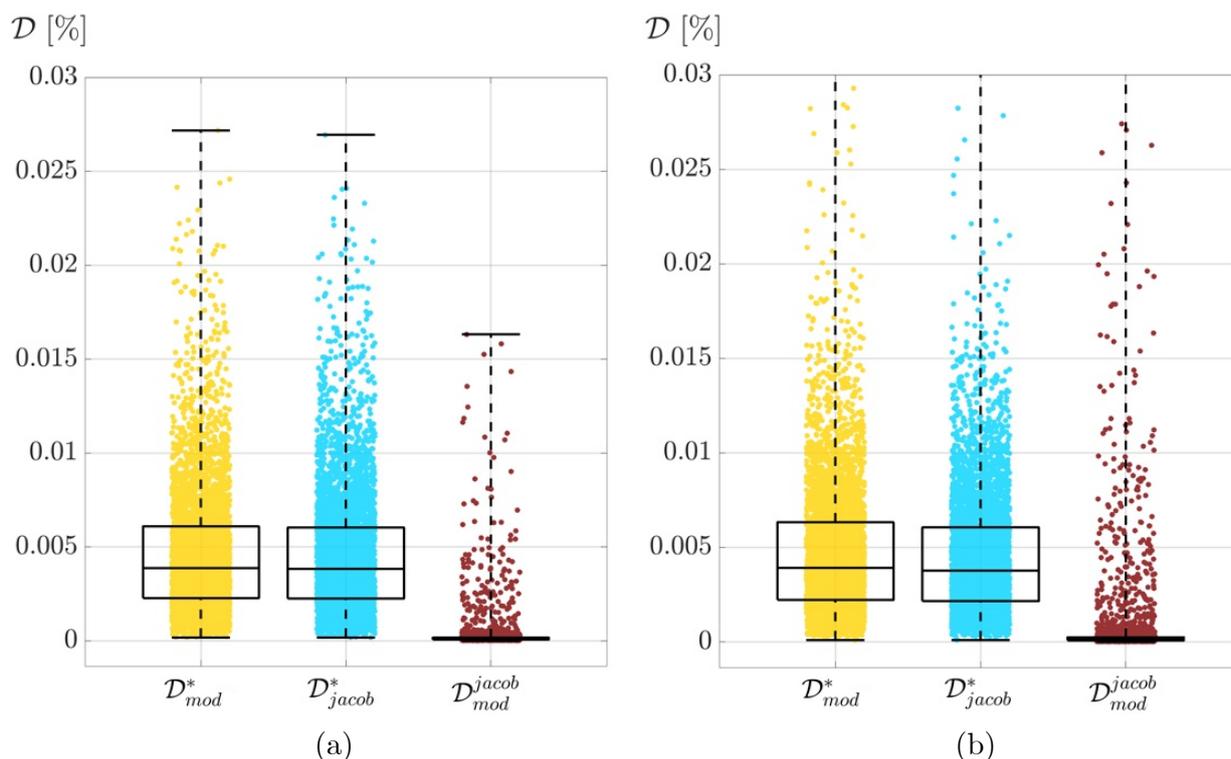


FIGURE 3.8 – Déformations relatives mesurées à partir des formes simulées grâce au modèle (\mathcal{D}_{mod}^*) et grâce aux matrices $C_{s_0}(s)$ et $J(s)$ calculées par la MPDBN (\mathcal{D}_{jacob}^*). La différence entre ces deux déformations est mesurée par $\mathcal{D}_{mod}^{jacob}$. Les résultats présentés ont été obtenus à partir de 5000 simulations de formes sans précharge (a) et avec précharge (b).

	Sans précharge			Avec précharge		
	\mathcal{D}_{mod}^* [%]	\mathcal{D}_{jacob}^* [%]	$\mathcal{D}_{mod}^{jacob}$ [%]	\mathcal{D}_{mod}^* [%]	\mathcal{D}_{jacob}^* [%]	$\mathcal{D}_{mod}^{jacob}$ [%]
Minimum	$1.7 \cdot 10^{-4}$	$1.7 \cdot 10^{-4}$	$7.5 \cdot 10^{-6}$	$8.5 \cdot 10^{-5}$	$1.0 \cdot 10^{-4}$	$8.7 \cdot 10^{-6}$
Médiane	$3.9 \cdot 10^{-3}$	$3.8 \cdot 10^{-3}$	$1.1 \cdot 10^{-4}$	$4.0 \cdot 10^{-3}$	$3.8 \cdot 10^{-3}$	$1.4 \cdot 10^{-4}$
Moyenne	$4.6 \cdot 10^{-3}$	$4.6 \cdot 10^{-3}$	$2.5 \cdot 10^{-4}$	$1.3 \cdot 10^{-2}$	$4.6 \cdot 10^{-3}$	$8.8 \cdot 10^{-3}$
75 ^{ième} centile	$6.1 \cdot 10^{-3}$	$6.0 \cdot 10^{-3}$	$1.6 \cdot 10^{-4}$	$6.3 \cdot 10^{-3}$	$6.1 \cdot 10^{-3}$	$2.5 \cdot 10^{-4}$
90 ^{ième} centile	$9.0 \cdot 10^{-3}$	$8.9 \cdot 10^{-3}$	$2.5 \cdot 10^{-4}$	$9.6 \cdot 10^{-3}$	$9.0 \cdot 10^{-3}$	$5.2 \cdot 10^{-4}$
99 ^{ième} centile	$1.7 \cdot 10^{-2}$	$1.7 \cdot 10^{-2}$	$4.2 \cdot 10^{-3}$	$3.1 \cdot 10^{-2}$	$1.5 \cdot 10^{-2}$	$2.7 \cdot 10^{-2}$
Maximum	$2.7 \cdot 10^{-2}$	$2.7 \cdot 10^{-2}$	$1.6 \cdot 10^{-2}$	$5.6 \cdot 10^0$	$3.6 \cdot 10^{-2}$	$5.6 \cdot 10^0$

TABLE 3.7 – Mesures statistiques extraites à partir des formes simulées grâce au modèle (\mathcal{D}_{mod}^*) et grâce aux matrices $C_{s_0}(s)$ et $J(s)$ calculées par la MPDBN (\mathcal{D}_{jacob}^*). La différence entre ces deux déformations est mesurée par $\mathcal{D}_{mod}^{jacob}$. Les résultats présentés ont été obtenus à partir de 5000 simulations pour des formes sans précharge (a) et avec précharge (b).

Le [tableau 3.7](#) nous montre que la différence relative entre les deux méthodes $\mathcal{D}_{mod}^{jacob}$ et au moins d'un ordre de grandeur moindre que les déformations \mathcal{D}_{mod}^* et \mathcal{D}_{jacob}^* pour 99% des simulations sans précharge et 90% des simulations avec précharge. Sachant que des travaux ont montré que les efforts externes peuvent tout aussi bien améliorer [[Peyron et al., 2022](#)] ou dégrader [[Ha et al., 2015](#)] la stabilité des RTC, le seuil de 90% dans le cas avec précharge est très prometteur. Cette analyse nous amène à la conclusion que la MPDBN permet d'estimer correctement la compliance $C_{s_0}(s)$ et la jacobienne $J(s)$ dans respectivement 99% et 90% des simulations sans et avec précharge.

3.3.4 Temps de calcul

Il a été montré dans la [sous-section 3.3.3](#) que les résultats issus de la MPDBN semblent corrects par rapport à ceux issus de la MDFPVL. Cependant, l'efficacité de la MPDBN reste à prouver et c'est pourquoi nous avons mené une étude pour comparer les temps de calculs de notre MPDBN par rapport à ceux des MDFPVL, MDFPVI et MPDHN. Dans la [sous-section 3.2.4](#), nous avons vu que notre MPDBN peut réduire de 49% le nombre d'opérations par rapport à la MPDHN. Nous avons également précisé que ces estimations ne prennent pas en compte toutes les opérations machines telles que les opérations logicielles (comme les appels de variables par exemple) qui sont difficiles à estimer. Le langage de programmation peut également influencer grandement le temps de calcul : Matlab, en plus d'être un logiciel, est un langage haut-niveau exécuté à la volée tandis que le langage C/C++ est un langage bas-niveau et compilé. Une implémentation en C/C++ permettra donc un temps d'exécution plus faible que pour une implémentation en Matlab mais nécessitera un temps de développement plus long. Il est toutefois possible de développer les méthodes de calculs de la compliance avec Matlab (pour faciliter le développement) puis de les convertir de Matlab vers C++ grâce à l'application *Matlab Coder* et des fonctions MEX pour réduire les temps d'exécution.

En utilisant ces outils, nous avons donc pu implémenter en C++ les 4 méthodes présentées dans la [sous-section 2.2.6](#). Puisque les temps de calcul seront estimés dans cette partie, on précisera que la machine utilisée dispose d'un processeur Apple M1 Pro de 10 cœurs et une fréquence de calcul de 3.22 GHz. Les méthodes implémentées ont ensuite été utilisées pour calculer la compliance $C_{s_0}(s)$, pour la configuration [RTC2-C1](#) et pour une discrétisation variant de $N_P = 30$ à $N_P = 100$ points. Il convient de préciser que la jacobienne $J(s)$ n'est pas calculée dans cette étude puisqu'elle n'était pas considérée dans les estimations des nombres d'opérations. Pour chaque discrétisation N_P , chaque programme a été exécuté 10 fois et le temps moyen a été calculé. La notation utilisée pour désigner les temps de calcul est analogue à celle du nombre d'opérations : T_{PVL}^{DF} désigne le temps de calcul pour l'exécution de la MDFPVL, T_{PVI}^{DF} pour la MDFPVI, T_{HN}^{PD} pour la MPDHN et T_{BN}^{PD} pour la MPDBN. De la même manière, on définira par (3.51) le gain de temps relatif permis grâce à la MPDBN par rapport à la MPDHN par l'équation :

$$G_T(\text{MPDBN}) = \frac{\|T_{HN}^{PD} - T_{BN}^{PD}\|}{\|T_{HN}^{PD}\|} \quad (3.51)$$

L'évolution des temps de calcul ainsi que du gain de temps permis par la MPDBN par rapport à la MPDHN sont présentés par la figure 3.9. On observe que notre MPDBN est systématiquement plus rapide que toutes les autres méthodes et que le gain de temps par rapport à la MPDHN augmente avec la finesse de la discrétisation, ce qui est en accord avec les calculs théoriques sur les nombres d'opérations présentés dans la sous-section 3.2.4. On constate également que la MPDBN permet de calculer complètement la compliance $C_{s_0}(s)$ à une fréquence de 100Hz pour $N_P = 100$ points et peut atteindre 500Hz en réduisant les points de discrétisation à $N_P = 30$. En fonction des applications envisagées, il faudra alors trouver un compromis entre la finesse de la discrétisation et la rapidité de calcul². Par exemple, un algorithme de planification de trajectoires pourra privilégier une discrétisation fine tandis qu'un algorithme de commande, nécessitant une fréquence de calcul élevée, pourra préférer une discrétisation plus large.

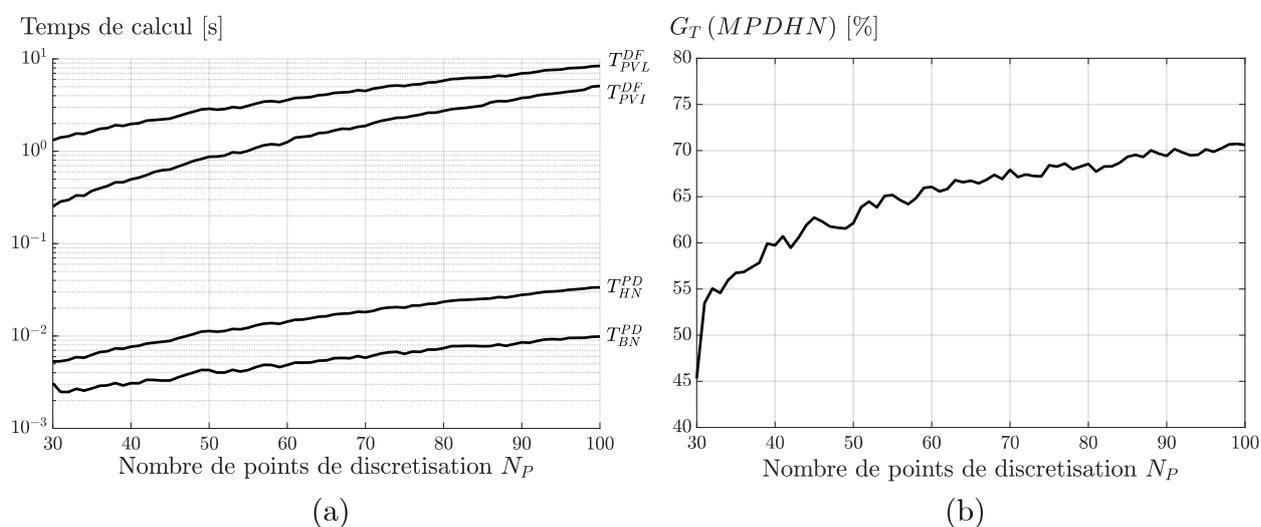


FIGURE 3.9 – Évolution du temps nécessaire au calcul de la compliance $C_{s_0}(s)$ en fonction du nombre de points de discrétisation N_P et pour les 4 méthodes présentées (a) et du gain de temps permis par la MPDBN par rapport à la méthode la plus efficace de la littérature, la MPDHN (b).

2. Il est d'ailleurs tout à fait envisageable de calculer le modèle cinéto-statique du robot et la matrice de compliance généralisée avec des discrétisations différentes en effectuant un ré-échantillonnage de l'abscisse curviligne du robot.

3.4 Discussions et perspectives

Nous avons vu dans ce chapitre la contribution centrale de cette thèse : la matrice de compliance généralisée (3.5) qui permet de quantifier complètement la compliance des robots continus flexibles. Afin d'utiliser cette nouvelle information pour diverses applications, il était nécessaire de disposer d'une méthode de calcul efficace. Les méthodes existantes jusqu'alors dans la littérature ne permettaient de calculer la compliance de façon optimisée que dans un cas non général [Rucker and Webster, 2011], ou pour une simple poutre de Cosserat et non un robot complet [Smoljkic et al., 2014]. Dans ce chapitre, nous avons vu comment tirer le meilleur parti des méthodes de la littérature, les adapter et les étendre, afin d'en développer une spécifiquement optimisée pour la compliance généralisée : la MPDBN. Les simulations réalisées ont montré que les résultats issus de la MPDBN sont semblables à ceux issus de la méthode de référence, la MDFPVL, tout en réduisant drastiquement le temps de calcul nécessaire. Ces résultats sont très prometteurs et amènent un certain nombre de perspectives.

3.4.1 Discussions sur la validation numérique

Premièrement, la validation présentée dans la sous-section 3.3.3 a été réalisée de manière à couvrir un grand nombre de scénarios aléatoires afin d'extraire des critères de performance représentatifs. Il faut toutefois rappeler que les résultats présentés par la figure 3.8 dépendent d'une certaine manière des caractéristiques du RTC2 simulé. En effet, la longueur du RTC, la taille de sa section et ses propriétés matérielles influencent les valeurs de la matrice de compliance généralisée. De plus, les paramètres qui ont été utilisés lors des simulations, listés dans les tableaux 3.5 et 3.6, impactent l'amplitude des déformations mesurées et donc également les résultats présentés. Ainsi, les valeurs de déplacement en fonction d'un niveau de force en Newtons sont à mettre en regard face à la raideur du robot considéré, et seront certainement application-dépendants en pratique.

3.4.2 Perspectives sur l'étude du point de contact

Il est également intéressant de noter l'effet du point d'application de la force sur les résultats. En effet, lorsqu'un robot continu est déformé par une force, on s'attend à observer des déformations d'amplitude croissante à mesure que l'intensité de la force augmente et à mesure que le point d'application s'éloigne de la base. Puisque la matrice de compliance généralisée est utilisée pour linéariser des déformations autour d'une forme donnée, on s'attend également à observer une erreur relative $\mathcal{D}_{mod}^{jacob}$ de plus en plus importante à mesure que l'amplitude des déformations augmente. Cependant, lors de la validation de la sous-section 3.3.3, les points d'application et les intensités des forces simulées ont été choisis aléatoirement. La figure 3.10a montre un ensemble de simulations réalisées sur le RTC2 pour clarifier la dépendance de l'erreur de modélisation issue de la compliance généralisée $\mathcal{D}_{mod}^{jacob}$ par rapport au point d'application s_0 et à l'intensité $\|\Delta f_0(s_0)\|$ de la force appliquée. Chaque point représente la moyenne de 50 simulations de forces appliquées selon des directions aléatoires. Comme attendu, on observe que l'erreur relative moyenne, issue

de la linéarisation des déformations par la matrice de compliance généralisée, augmente à mesure que le point d'application s'éloigne de la base du RTC et à mesure que l'intensité de la force appliquée devient importante. L'erreur $\mathcal{D}_{mod}^{jacob}$ semble augmenter linéairement par rapport à $\|\Delta f_0(s_0)\|$ sur $\|\Delta f_0(s_0)\| \in [10^{-3}, 1]$ N puis de manière plus importante lorsque $\|\Delta f_0(s_0)\| \geq 1$ N. De façon générale, la [figure 3.10](#) nous indique que la matrice de compliance généralisée permet d'estimer, par linéarisation, les déformations du [RTC2](#) avec moins de 1% d'erreur tant que l'intensité des forces reste inférieure à 10 N. Dans certains cas (voir les exemples (b) et (c) de formes reportés en [figure 3.10](#)) des erreurs faibles sont observées pour des forces importantes, jusqu'à 100 N lorsque les forces sont appliquées sur la première moitié du robot, ce qui présente un potentiel intéressant au niveau applicatif. Encore une fois, ces résultats sont robot- et configuration-dépendants et une étude plus complète serait nécessaire.

3.4.3 Perspectives sur le cas multi-forces

Bien que ces résultats soient très satisfaisants, dans un scénario chirurgical un robot continu sera potentiellement amené à subir de multiples interactions simultanément. Lors de la validation présentée dans la [sous-section 3.3.3](#), des variations d'actionnement, d'effort en bout et le long du robot ont été appliquées simultanément. Cependant le cas avec plusieurs efforts le long du robot n'a pas été traité. La [figure 3.11](#) montre des simulations quantifiant l'erreur de linéarisation $\mathcal{D}_{mod}^{jacob}$ en fonction du nombre de forces N_f appliquées sur le [RTC2](#), pour des valeurs d'intensité de force $\|\Delta f_0(s_0)\|$ variant de 0.01 N à 100 N. Pour ce scénario avec plusieurs forces, la déformation globale du RTC est estimée en superposant les déformations générées individuellement par chaque force grâce à l'équation [\(3.46\)](#). Comme précédemment, chaque point représente la moyenne de 50 essais avec des directions de force aléatoires. Comme attendu, la [figure 3.11](#) montre que l'erreur de linéarisation augmente avec le nombre de forces appliquées simultanément N_f . On observe également que l'erreur relative moyenne augmente avec l'intensité des forces $\|\Delta f_0(s_0)\|$, ce qui a déjà été observé précédemment avec des forces uniques. De manière plus intéressante, on observe que l'erreur $\mathcal{D}_{mod}^{jacob}$ augmente plus vite avec l'intensité des forces $\|\Delta f_0(s_0)\|$ qu'avec le nombre de forces N_f . En effet, l'erreur de linéarisation reste inférieure à 1% pour des scénarios où 100 forces de moins de 1 N sont appliquées simultanément et jusqu'à 20 forces de moins de 10 N. Si une étude complète serait certainement nécessaire pour le confirmer, ce résultat semble montrer que la matrice de compliance généralisée peut être utilisée pour des scénarios complexes impliquant un grand nombre de forces de contact.

3.4.4 Perspectives sur le modèle de compliance généralisée

Les résultats et discussions précédents indiquent que la méthode proposée dans ce chapitre a un potentiel intéressant pour de nombreuses applications en conception de robots continus [[Ha et al., 2016](#)] ou en commande [[Chikhaoui and Rosa, 2022](#)]. Il est intéressant de relever que si les calculs détaillés et validations ont été réalisés pour un RTC, le concept de compliance généralisée ainsi que les développements mathématiques de la MPDBN sont génériques et peuvent s'appliquer à tout type de robot continu dont le mo-

dèle peut s'écrire sous la forme canonique $\dot{y}(s) = \mathcal{F}(y, s)$. Ceci inclut les robots continus multi-backbone [Chen et al., 2021a], les robots parallèles continus [Orekhov et al., 2017], ou encore les robots continus à actionnement pneumatique [Till et al., 2018]. Une exception notable concerne les RC. En effet, la modélisation Newtonienne de ce type de robots conduit à une équation différentielle dont la forme est $M(y, s)\dot{y}(s) = \mathcal{F}(y, s)$, avec $M(y, s)$ une matrice de masse dépendant de l'état. La résolution de ce type de modèle est typiquement faite par inversion numérique de la matrice $M(y, s)$ [Rucker and Webster III, 2011, Till et al., 2018], ce qui rend une application directe difficile. Une solution pour appliquer la MPDBN consisterait à intégrer $M^{-1}(y, s)$ dans le vecteur d'état, en exploitant la relation $\frac{\partial}{\partial x}(M^{-1}) = -M^{-1}\frac{\partial M}{\partial x}M^{-1}$. L'impact de ces calculs sur la stabilité de la MPDBN n'est toutefois pas clairement établi.

Parallèlement à cette extension à d'autres types de robots, une extension de la MPDBN à d'autres algorithmes de résolution des modèles Cosserat de robot continu pourrait également être intéressante. En effet, les concepts développés dans cette thèse se basent sur un modèle newtonien, mais d'autres formalismes exploitant un modèle lagrangien ont été développés ces dernières années notamment avec les travaux de [Boyer et al., 2020, Childs and Rucker, 2023, Tummers et al., 2024]. Ces modèles opèrent une réduction de la cinématique par projection sur une base de fonctions continues par morceaux [Pourafzal et al., 2021], linéaires par morceaux [Li et al., 2023], ou polynomiales [Boyer et al., 2020] pour ensuite appliquer un principe de la mécanique tel que les travaux virtuels par exemple. Dans [Tummers et al., 2023] une méthode de propagation de dérivées est notamment exploitée pour calculer la jacobienne du problème d'optimisation et accélérer la résolution du modèle lagrangien, donnant une indication sur la possibilité d'extension de notre méthode à ce type de modèles. Par ailleurs, des recherches récentes ont montré qu'il était possible d'interfacer des modèles de Cosserat avec des simulations par éléments finis, en reformulant les efforts d'interaction dans un repère de référence global [Adagolodjo et al., 2021]. Notre modèle de compliance généralisée pourrait ici ouvrir des perspectives intéressantes pour accélérer la résolution des modèles de robots continus en interaction avec des structures complexes simulées par éléments finis.

Enfin, il ne faut pas oublier que les concepts développés dans ce chapitre restent avant tout des modèles, dont la validité dépend des hypothèses simplificatrices présentées dans le chapitre 2. Bien que le modèle cinétostatique soit plutôt réaliste avec moins de 3 mm d'erreur sur la position du bout du robot [Lock et al., 2010, Rucker et al., 2010], il n'est pas évident d'anticiper comment ces hypothèses de modélisation se répercuteraient sur le réalisme du modèle de compliance développé dans ce chapitre.

3.4.5 Perspectives générales sur la compliance généralisée

La matrice de compliance généralisée, définie par l'équation (3.5) et calculée grâce à la MPDBN présentée à la sous-section 3.2.4, est un outil très prometteur pour quantifier et utiliser l'information de compliance des robots continus et spécifiquement des RTC. Nous verrons dans le chapitre suivant comment cet outil peut être exploité pour une application pratique : l'estimation des forces agissant sur un robot continu.

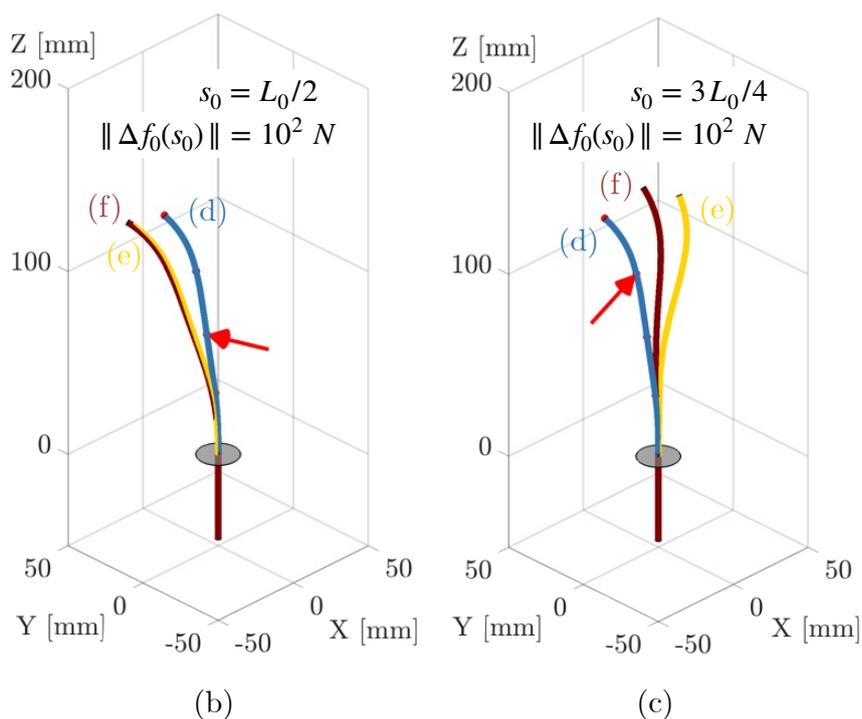
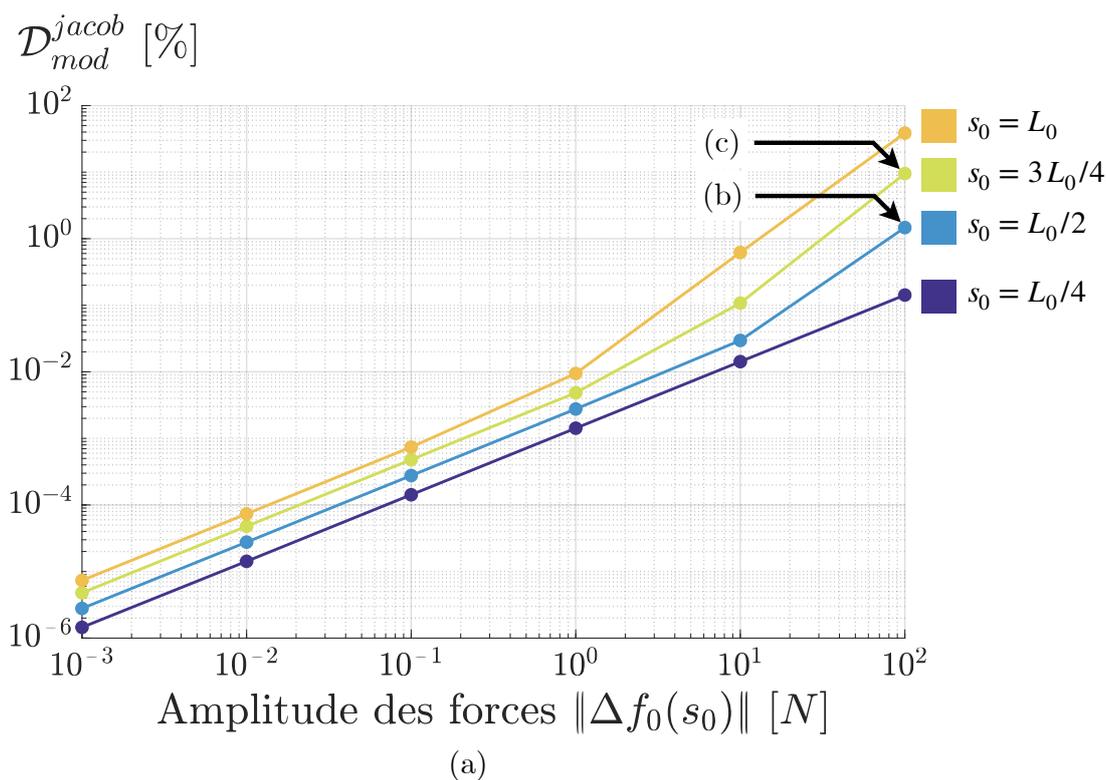


FIGURE 3.10 – (a) Visualisation des erreurs moyennes de linéarisation $\mathcal{D}_{mod}^{jacob}$ des déformations du RTC2 en fonction du point d'application s_0 des forces et de leurs intensités $\|\Delta f_0(s_0)\|$. Les graphiques (b) et (c) illustrent respectivement les formes obtenues lorsque des forces de 100 N sont appliquées en $s_0 = \frac{L_0}{2}$ et $s_0 = \frac{3L_0}{4}$. On y retrouve la forme initiale $p^*(s)$ (d), la forme déformée estimée grâce au modèle de forme $p_{mod}(s)$ (e) et la forme déformée estimée par linéarisation grâce à la matrice de compliance généralisée $p_{jacob}(s)$ (f).

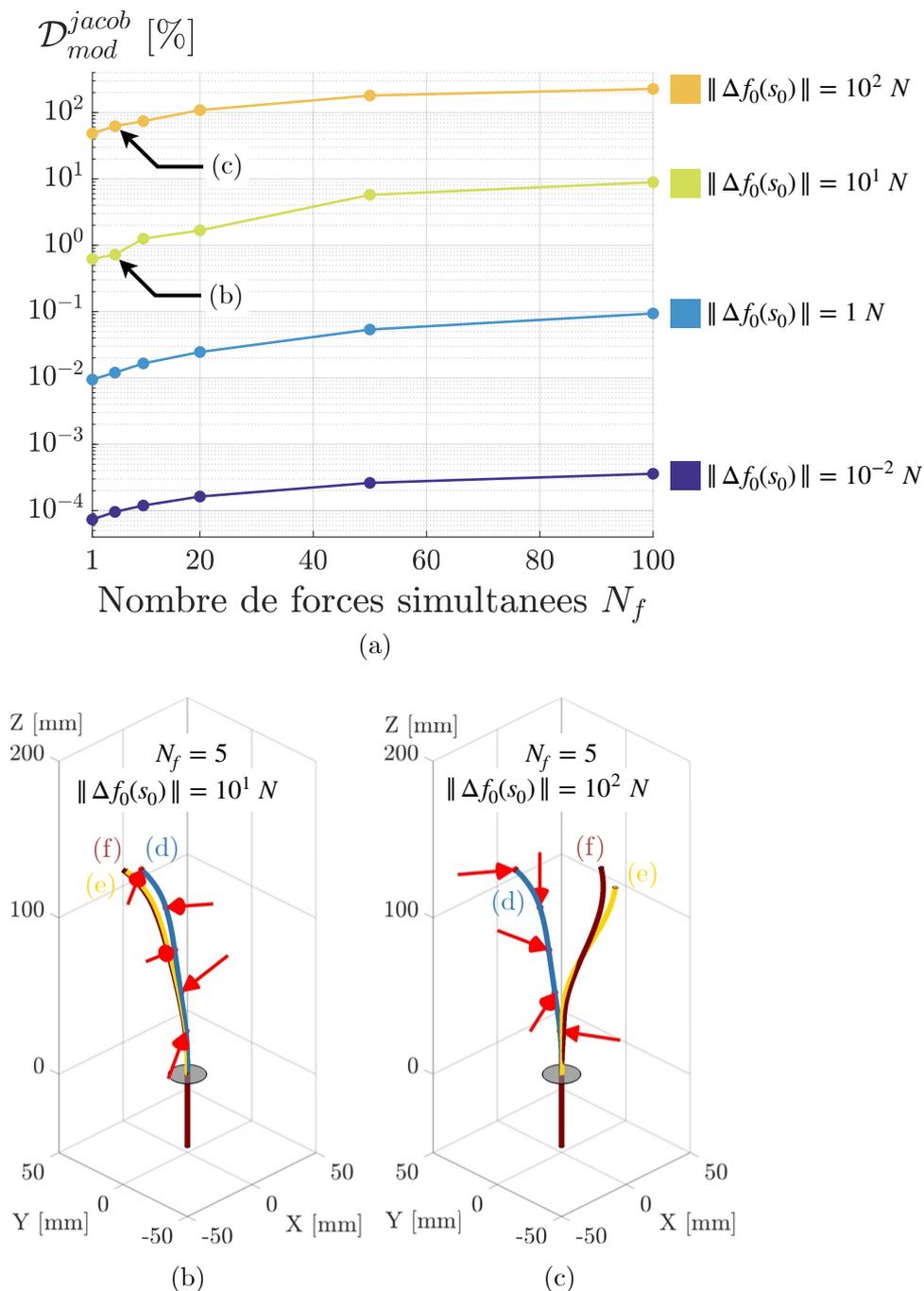


FIGURE 3.11 – (a) Visualisation des erreurs moyennes de linéarisation $\mathcal{D}_{mod}^{jacob}$ des déformations du RTC2 en fonction du nombre de forces appliquées simultanément N_f et de leurs intensités $\|\Delta f_0(s_0)\|$. Les graphiques (b) et (c) illustrent respectivement les formes obtenues lorsque 5 forces de 10 N et 100 N sont appliquées simultanément sur le RTC. On y retrouve la forme initiale $p^*(s)$ (d), la forme déformée estimée grâce au modèle de forme $p_{mod}(s)$ (e) et la forme déformée estimée par linéarisation grâce à la matrice de compliance généralisée $p_{jacob}(s)$ (f).

Chapitre 4

Estimation d'efforts externes basée compliance

Nous avons vu dans le [chapitre 2](#), et en particulier dans l'équation (2.22), que les efforts appliqués apparaissent comme des entrées du modèle cinétostatique. Pour modéliser au plus juste la forme des robots continus, il devient donc nécessaire de mesurer ces efforts ou à défaut de les estimer. Cette mesure d'efforts ne vise pas uniquement à améliorer le modèle de la forme des robots continus, mais présente également un intérêt pour de nombreuses autres applications. Lorsque l'on a connaissance de l'environnement dans lequel le robot est déployé et de ses propriétés mécaniques, il devient par exemple possible d'utiliser ces forces de contact pour atteindre de nouvelles formes, inaccessibles autrement [[Rao et al., 2024](#)]. D'autres travaux visent à exploiter ces mesures de force pour aider le chirurgien dans son intervention en détectant automatiquement des collisions [[Lin et al., 2019](#), [Frazelle et al., 2020](#)], en utilisant des guides virtuels pour éviter que le robot ne s'éloigne trop d'une trajectoire donnée [[Xie et al., 2022](#)] ou encore en déployant automatiquement le robot grâce à ces informations [[Goldman et al., 2014](#), [Fagogenis et al., 2019](#)].

Dans ce chapitre, nous allons nous intéresser aux solutions permettant de mesurer ou d'estimer ces efforts. Un état de l'art des solutions existantes sera tout d'abord présenté dans la [section 4.1](#). A l'issue de cette présentation, la [section 4.2](#) expliquera comment la matrice de compliance généralisée, présentée dans le [chapitre 3](#), peut être utilisée pour formuler le problème d'estimation des efforts d'interaction en les reliant aux déformations du robot. Finalement, la méthode d'estimation des efforts sera appliquée au cas des RTC dans la [section 4.3](#), avant de discuter les éléments présentés dans la [section 4.4](#).

Les nouvelles variables introduites dans ce chapitre sont disponibles en annexes dans le [tableau 5.12](#).

4.1 État de l'art

Lorsque l'on étudie les solutions d'estimation d'efforts, on peut tout d'abord penser aux efforts exercés par l'effecteur du robot. La grande majorité des capteurs de force commercialisés proposent un compromis : réduire la taille du capteur implique de réduire le nombre de composantes (*axes*) d'efforts mesurées, i.e. forces et couples dans le cas général. La miniaturisation des capteurs de force est une thématique de recherche à part entière, portée à la fois par les industriels et les laboratoires de recherche. On pourra notamment mentionner le capteur 3 axes de 1 *cm* de largeur développé par [Puangmali et al., 2011] et testé dans le cadre d'une palpation ex-vivo ou encore le capteur multi-directionnel de moins de 1 mm développé par [Lou et al., 2010]. Lorsque l'on cherche à instrumenter des robots continus, l'estimation d'efforts doit être réalisée sur toute la longueur du robot et pas seulement au bout. Une approche consiste alors à intégrer plusieurs capteurs de forces conçus sur mesure à l'intérieur même du robot [Noh et al., 2014]. Dans le cas d'un coloscope, des travaux [Korn et al., 2024] présentent des capteurs souples, fixés sur la périphérie du corps, pour détecter et quantifier des efforts de contact. Puisque la détermination des efforts le long d'un robot implique notamment de localiser les points de contact, il devient alors nécessaire de multiplier les capteurs utilisés. Bien que de nouvelles technologies de capteurs de force soient en cours de développement, l'intégration d'un grand nombre de ces capteurs dans un robot continu pose de nombreux défis techniques et entraîne inévitablement une hausse du coût de fabrication du robot.

Cette limite nous amène à considérer une approche différente, en estimant les efforts d'interaction sans instrumenter directement le robot avec des capteurs de force. L'approche consiste à utiliser la mesure d'une grandeur qui varie selon les efforts appliqués sur le robot, et dont la mesure est plus simple à réaliser. Des modèles sont ensuite utilisés pour relier les mesures de cette grandeur aux efforts appliqués sur le robot. Ce type d'approche a été explorée par Alian et al. [Alian et al., 2023] en utilisant un fluide conducteur pour actionner un robot hydraulique ainsi que des électrodes, fixées à la base du robot, pour mesurer l'impédance du fluide. Lorsqu'un effort est appliqué sur le robot, celui-ci se déforme et la répartition du fluide à l'intérieur du robot s'en trouve modifiée. Ceci fait alors varier l'impédance mesurée. Sans connaissance a priori du modèle reliant l'effort et l'impédance électrique du fluide, un modèle expérimental est alors utilisé.

Nous avons vu dans le chapitre 2 qu'un grand nombre de travaux ont d'ores et déjà été développés sur la modélisation cinétostatique des robots continus. On peut donc envisager d'utiliser ce modèle comme intermédiaire : il est par exemple possible d'estimer la force appliquée au bout d'un RC en utilisant une mesure de la tension d'actionnement des câbles [Du et al., 2024]. Il est intéressant de remarquer que la plupart des capteurs de force utilisent en pratique une mesure de déformation comme grandeur physique intermédiaire à l'estimation d'efforts. C'est le cas par exemple des capteurs utilisant des jauges de contrainte, des capteurs capacitifs ou encore des capteurs piézoélectriques. On peut donc imaginer appliquer un principe équivalent aux robots continus pour ramener le problème d'estimation d'efforts à une estimation de leur forme.

La suite de notre étude se focalise ainsi sur les solutions permettant d'estimer les efforts d'interaction en mesurant les déformations du robot (système actionné) ou d'une poutre de Cosserat le cas échéant (système non actionné). Cette thématique se développe dans la communauté scientifique, portée notamment par la commercialisation de Réseaux de Fibres de Bragg (RFB) qui, lorsqu'ils sont intégrés dans un endoscope, permettent d'en mesurer la courbure pour finalement en estimer la forme [Ryu and Dupont, 2014], [Wang et al., 2023a]. Une autre approche pour mesurer la forme du robot consiste à reconstruire la forme en 3D grâce à plusieurs images 2D du robot. Cette approche, que nous allons adopter, sera présentée dans la [sous-section 4.3.2](#).

En admettant que la forme du robot puisse être mesurée, il devient possible de formuler le problème d'estimation d'efforts en bout comme un problème d'optimisation dans le cas d'un RC [Xu and Simaan, 2008] ou d'un problème d'inversion de matrice dans le cas d'une poutre de Cosserat [Diezinger et al., 2022]. Lorsque l'on généralise l'estimation des efforts sur toute la longueur du robot, nous ne disposons plus de connaissance a priori des points sur lesquels les interactions ont lieu. Pour répondre à cette problématique, les points d'applications potentiels ont été inclus dans les problèmes d'optimisation formulés pour des poutres de Cosserat [Aloi and Rucker, 2019, Al-Ahmad et al., 2021, Qiao et al., 2021] et pour des RC [Aloi et al., 2022]. Ces problèmes d'optimisation sont complexes, nécessitent de nombreuses itérations pour converger et demandent la résolution du modèle cinéto-statique à chaque itération. Par conséquent, l'estimation des efforts requiert d'importantes ressources de calcul et le temps de résolution peut être devenir important. Plus récemment, les travaux de [Ferguson et al., 2024] ont permis de formuler un problème unique incluant à la fois la modélisation de la forme et l'estimation des efforts. Cette approche originale suppose elle aussi une mesure de la forme réelle du robot et considère les efforts appliqués sur le robot comme des inconnues supplémentaires à estimer lors de la résolution du modèle cinéto-statique.

Puisque nous disposons désormais d'un nouvel outil de modélisation, la matrice de compliance généralisée, reliant la forme du robot aux efforts d'interaction, on peut tout naturellement envisager d'utiliser cette matrice pour étudier le problème d'estimation d'efforts à partir de la mesure de la forme du robot.

4.2 Estimation d'efforts grâce à une mesure de forme

Nous présenterons dans la [sous-section 4.2.1](#) comment la matrice de compliance généralisée peut permettre de représenter le problème d'estimation sous une nouvelle forme, linéaire et géométriquement exacte. Les caractéristiques de cette nouvelle formulation seront étudiées dans la [sous-section 4.2.2](#) puis une méthode générale de résolution sera exposée dans la [sous-section 4.2.3](#). Cette méthode sera ensuite appliquée dans la [sous-section 4.2.4](#) au cas particulier de l'estimation de force à partir d'une mesure de position, sans considérer directement ni les couples ni l'orientation du robot. Finalement, nous verrons dans la [sous-section 4.2.5](#) comment la matrice compliance généralisée peut être utilisée pour quantifier l'incertitude de l'estimation de force en la corrélant à l'incertitude de mesure de la forme du robot.

4.2.1 Compliance généralisée et estimation d'efforts

Nous avons vu que la définition de la matrice de compliance généralisée (3.5) permet de relier une variation de pose du robot $\partial T_0^\vee(s)$ à une variation d'effort $\partial w_0(s_0)$. On supposera par la suite que ni les propriétés du robot ni son actionnement ne varient lors de l'estimation d'effort. Ainsi toute déformation de la pose du robot, notée $\Delta T_0^\vee(s, s_0)$, est forcément générée par une variation d'effort, notée $\Delta w_0(s_0)$. Sous ces conditions et grâce à la définition de la matrice de compliance généralisée (3.5), on obtient :

$$\forall (s, s_0) \in [0, L_0]^2, \Delta T_0^\vee(s, s_0) \approx C_{s_0}(s) \Delta w_0(s_0) \quad (4.1)$$

Afin d'éviter toute ambiguïté, nous désignerons de deux façons différentes une déformation de la forme du robot :

- $\Delta T_0^\vee(s, s_0)$ pour une déformation estimée grâce à la compliance généralisée et associée à unique effort $\Delta w_0(s_0)$ d'abscisse s_0 connue.
- $\Delta T_0^\vee(s)$ pour une déformation physique supposée mesurable et associée potentiellement à plusieurs efforts $\Delta w_0(s_0)$ d'abscisses s_0 différentes et inconnues.

Dans le cas le plus général, des efforts peuvent être appliqués sur toute la longueur du robot $[0, L_0]$ et chaque déformation $\Delta T_0^\vee(s, s_0)$ va contribuer à former la déformation globale $\Delta T_0^\vee(s)$. En utilisant le principe de superposition, on peut alors voir la déformation $\Delta T_0^\vee(s)$ comme la somme de toutes les petites déformations $\Delta T_0^\vee(s, s_0)$:

$$\forall s \in [0, L_0], \Delta T_0^\vee(s) \approx \sum_{s_0 \in [0, L_0]} \Delta T_0^\vee(s, s_0) \quad (4.2)$$

et en injectant l'équation (4.1), on obtient :

$$\forall s \in [0, L_0], \Delta T_0^\vee(s) \approx \sum_{s_0 \in [0, L_0]} C_{s_0}(s) \Delta w_0(s_0) \quad (4.3)$$

L'équation (4.3), désignée comme une formulation analytique du *problème direct de déformation* du robot, peut également mise sous la forme matricielle (4.4) qui est équivalente à (4.3) :

$$\begin{bmatrix} \Delta T_0^\vee(0) \\ \vdots \\ \Delta T_0^\vee(s) \\ \vdots \\ \Delta T_0^\vee(L_0) \end{bmatrix} \approx \begin{bmatrix} C_0(0) & \dots & C_{s_0}(0) & \dots & C_{L_0}(0) \\ \vdots & & \vdots & & \vdots \\ C_0(s) & \dots & C_{s_0}(s) & \dots & C_{L_0}(s) \\ \vdots & & \vdots & & \vdots \\ C_0(L_0) & \dots & C_{s_0}(L_0) & \dots & C_{L_0}(L_0) \end{bmatrix} \begin{bmatrix} \Delta w_0(0) \\ \vdots \\ \Delta w_0(s_0) \\ \vdots \\ \Delta w_0(L_0) \end{bmatrix} \quad (4.4)$$

que l'on désignera de manière plus compacte par :

$$\Delta \mathcal{T}_0^\vee \approx \mathcal{C} \Delta \mathcal{W}_0 \quad (4.5)$$

On pourra remarquer que, par construction, $\Delta \mathcal{W}_0$ et $\Delta \mathcal{T}_0^\vee$ sont des vecteurs de taille $6N_P \times 1$. A partir de (4.5), on comprend que si la déformation $\Delta \mathcal{T}_0^\vee$ est mesurable et que la matrice \mathcal{C} est inversible, alors il serait possible de calculer en une fois l'intégralité des variations d'efforts $\Delta \mathcal{W}_0$ appliqués sur le robot :

$$\Delta \mathcal{W}_0 \approx \mathcal{C}^{-1} \Delta \mathcal{T}_0^\vee \quad (4.6)$$

L'équation (4.6) sera désignée comme une formulation matricielle du *problème inverse de déformation*. Les deux formulations, directe (4.5) et inverse (4.6), sont géométriquement exactes lorsque $\Delta \rightarrow 0$ et ne requièrent pas d'hypothèses spécifiques autres que celles du modèle cinétostatique. Contrairement aux méthodes existantes, le modèle cinétostatique du robot n'intervient pas car toute l'information nécessaire pour résoudre le problème a d'ores et déjà été extraite et est contenue dans la matrice de compliance généralisée $C_{s_0}(s)$. Avant de pouvoir utiliser la formulation (4.6) pour résoudre notre problème d'estimation d'efforts, il convient toutefois de réaliser une étude sur le conditionnement de \mathcal{C} afin de vérifier son inversibilité.

4.2.2 Conditionnement du problème

En utilisant la fonction *cond* de Matlab, nous obtenons la valeur de conditionnement $\text{cond}(\mathcal{C}) \approx 10^{23}$ pour la configuration **RTC1-C1** ce qui ne permet donc pas de résoudre le problème inverse (4.6) par inversion de la matrice \mathcal{C} sans générer d'erreurs numériques très importantes. Pour mieux comprendre l'origine de ce problème, nous avons cartographié le conditionnement de $C_{s_0}(s)$ pour $(s, s_0) \in [0, L_0]^2$. Les résultats obtenus, présentés par la [figure 4.1](#), mettent en lumière l'existence de deux asymptotes (4.7) :

$$\text{cond}(C_{s_0}(s)) \xrightarrow{s_0 \rightarrow 0} +\infty \quad (4.7a)$$

$$\text{cond}(C_{s_0}(s)) \xrightarrow{s \rightarrow 0} +\infty \quad (4.7b)$$

On observe également que le conditionnement s'améliore, c'est-à-dire qu'il décroît, lorsque l'on s'éloigne des bords ($s \rightarrow 0$) et ($s_0 \rightarrow 0$) :

$$\frac{\text{cond}(C_{s_0=90}(s=90))}{\text{cond}(C_{s_0=10}(s=10))} \approx \frac{10^5}{10^{15}} = 10^{-10} \quad (4.8)$$

où les abscisses s et s_0 sont données en mm.

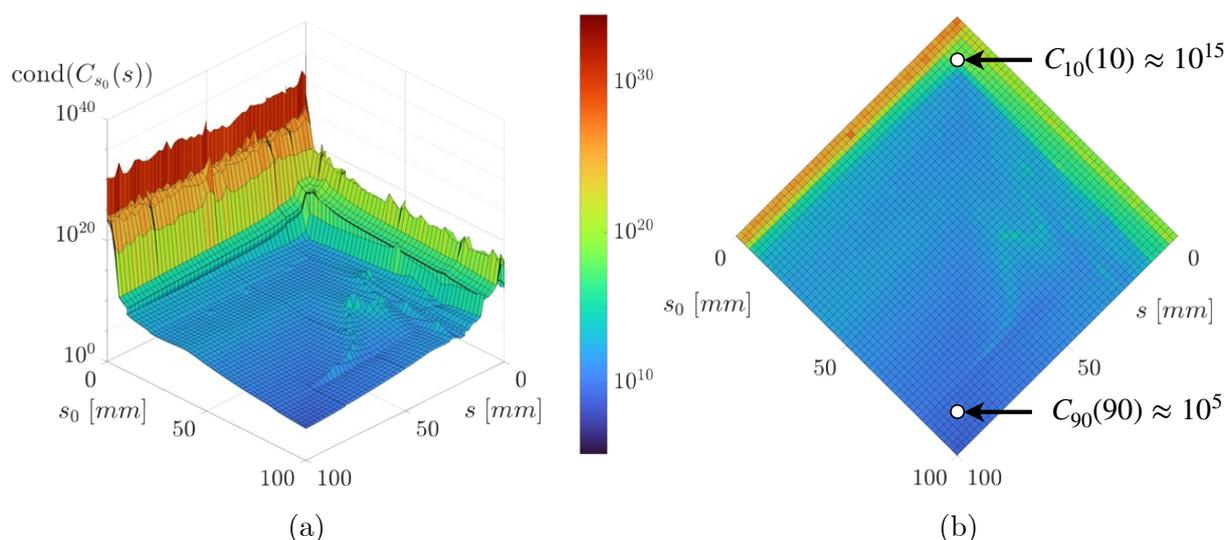


FIGURE 4.1 – Vue en 3D (a) et vue de dessus (b) de la cartographie du conditionnement de la matrice de compliance généralisée $C_{s_0}(s)$ en fonction de s et s_0 , réalisée pour la configuration RTC1-C1.

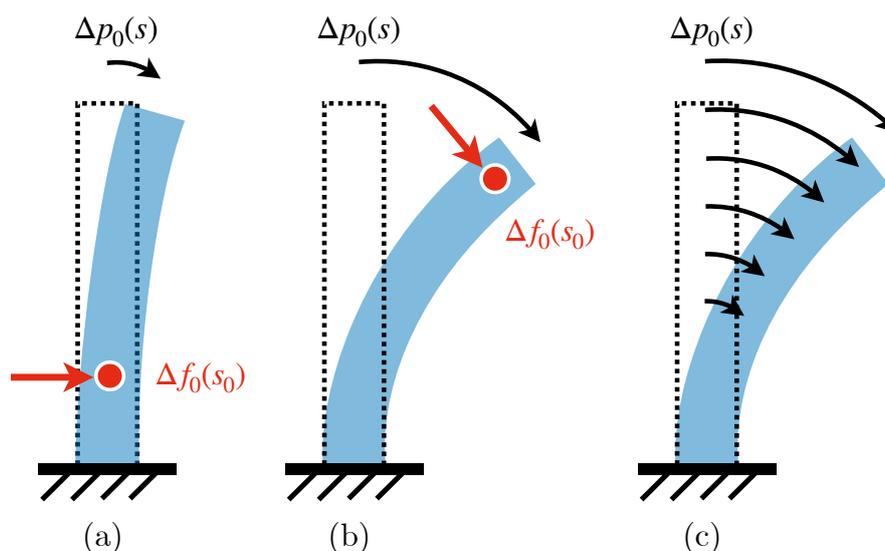


FIGURE 4.2 – Illustration pour une poutre encastée-libre du phénomène de bras de levier lorsqu'une force est appliquée à sa base (a) ou à son extrémité libre (b) ainsi que de son champ de déformation (c).

Ces phénomènes de bord, observés pour les RTC, se généralisent à tous les systèmes mécaniques déformables pouvant être assimilés à des poutres encastées-libres. Les schémas (a) et (b) de la [figure 4.2](#) illustrent la notion de *bras de levier* pour une force $\Delta f_0(s_0)$ appliquée perpendiculairement sur le robot. Lorsque la force est appliquée à proximité de la base, cas (a), la déformation globale de la poutre $\Delta p_0(s)$ sera moins importante que lorsque la force est appliquée vers l'extrémité, cas (b). Dans le cas (a), le bras de levier, c'est-à-dire la distance entre le point d'application de la force et la base, est plus

petit que dans le cas (b). Plus le bras de levier est important, plus le moment généré par $\Delta f_0(s_0)$ sera important. Cette notion de bras de levier est à l'origine de la limite observée à l'équation (4.7a). Le schéma (c) de la figure 4.2 représente quant à lui le *champ de déformation* de la poutre. Dans ce cas, l'intensité du moment fléchissant induit par $\Delta f_0(s_0)$ va croître à mesure que le point considéré s'éloigne de la base encastree et donc plus la déformation $\Delta p_0(s)$ sera importante. C'est ce champ de déformation qui est à l'origine de la limite observée à l'équation (4.7b). En conclusion, les phénomènes de bord (4.7) font partie intégrante du comportement en déformation du robot et le conditionnement ne peut donc pas être amélioré sans hypothèse simplificatrice supplémentaire.

4.2.3 Formulation d'un problème d'optimisation

Pour résoudre le problème inverse (4.6) sans inverser directement la matrice \mathcal{C} , nous proposons de formuler un problème d'optimisation à partir de la formulation directe (4.5). L'idée est de calculer par optimisation une estimation de la variation d'effort $\widetilde{\Delta \mathcal{W}}_0$ qui génère une déformation $\mathcal{C} \cdot \widetilde{\Delta \mathcal{W}}_0$ semblable à la déformation mesurée $\Delta \mathcal{T}_0^\vee$:

$$\widetilde{\Delta \mathcal{W}}_0 \approx \underset{\Delta \mathcal{W}_0}{\operatorname{argmin}} \left(\|\Delta \mathcal{T}_0^\vee - \mathcal{C} \Delta \mathcal{W}_0\|^2 \right) \quad (4.9)$$

où le symbole $\tilde{\bullet}$ indique une estimation par optimisation.

Appliquer directement la fonction d'optimisation (4.9) reviendrait à optimiser un total de $6 N_P$ variables, où N_P est le nombre de points de discrétisation du robot qui peut facilement atteindre $N_P \geq 100$. Afin de limiter les ressources de calcul nécessaires pour résoudre le problème d'estimation, nous proposons par la suite de réduire la taille du problème.

4.2.4 Étude du problème force-position

Jusqu'à maintenant nous avons formulé le problème d'estimation des efforts dans sa globalité or, en pratique, il est peu probable qu'un couple ne s'applique le long du robot, extrémité non incluse. De plus, la plupart des systèmes envisagés pour mesurer la forme des robots ne fournissent que des positions et non des orientations. Pour ces raisons, nous proposons ici de spécialiser notre étude dans le cas où l'on souhaite estimer des forces, et non des couples, grâce à une mesure de position, et non d'orientation. Nous utiliserons donc par la suite une sous-définition de la matrice de compliance généralisée, notée $C_{s_0}^*(s)$, qui est la sous-partie de la matrice de compliance généralisée originale (3.5) associée aux positions et aux forces :

$$\forall (s, s_0) \in [0, L_0]^2, C_{s_0}^*(s) = \frac{\partial p_0}{\partial f_0(s_0)}(s) \quad (4.10)$$

Les formulations analytiques (4.3) et matricielles (4.4) du problème direct de déformation du robot peuvent alors être adaptée respectivement selon (4.11) et (4.12).

$$\forall s \in [0, L_0], \Delta p_0(s) \approx \sum_{s_0 \in [0, L_0]} C_{s_0}^*(s) \Delta f_0(s_0) \quad (4.11)$$

$$\begin{bmatrix} \Delta p_0(0) \\ \vdots \\ \Delta p_0(s) \\ \vdots \\ \Delta p_0(L_0) \end{bmatrix} \approx \begin{bmatrix} C_0^*(0) & \dots & C_{s_0}^*(0) & \dots & C_{L_0}^*(0) \\ \vdots & & \vdots & & \vdots \\ C_0^*(s) & \dots & C_{s_0}^*(s) & \dots & C_{L_0}^*(s) \\ \vdots & & \vdots & & \vdots \\ C_0^*(L_0) & \dots & C_{s_0}^*(L_0) & \dots & C_{L_0}^*(L_0) \end{bmatrix} \begin{bmatrix} \Delta f_0(0) \\ \vdots \\ \Delta f_0(s_0) \\ \vdots \\ \Delta f_0(L_0) \end{bmatrix} \quad (4.12)$$

On remarquera que la taille du problème matriciel est ainsi réduite, passant de $(6N_P \times 6N_P)$ pour (4.4) à $(3N_P \times 3N_P)$ pour (4.12). Comme présenté dans la section 4.1, le problème d'estimation de forces grâce à une mesure de position a déjà été étudié à partir du modèle géométrique du robot. Dans les travaux [Al-Ahmad et al., 2021, Aloï et al., 2022, Ferguson et al., 2024], une hypothèse sur les forces appliquées, que l'on désignera par **(H1)**, est largement utilisée. Elle consiste à négliger les composantes axiales des forces de contact le long du robot, ce qui se traduit par :

$$\forall s_0 \in [0, L_0[, (R_0(s_0) \Delta f_0(s_0))|_z \approx 0 \quad (4.13)$$

[Al-Ahmad et al., 2021] considèrent que la déformation générée par une force axiale est négligeable puisque l'élongation et la compression du robot sont négligées. Cet argument, valable pour une poutre parfaitement rectiligne, devient discutable lorsque le robot prend des formes complexes et tri-dimensionnelles. [Aloï et al., 2022] ainsi que [Ferguson et al., 2024] considèrent quant à eux que les forces axiales sont négligeables car il n'y a que très peu de frottement lors des interactions du robot avec son environnement. Lors d'une intervention médicale, les contacts avec des tissus biologiques sont naturellement lubrifiés ce qui limite l'intensité de la composante axiale de la force. Toutefois, ces petits efforts axiaux peuvent s'accumuler sur toute la longueur du robot et il n'est pas trivial de prédire l'impact que cela aurait sur la déformation globale du robot. De plus, cette hypothèse devient discutable lorsque le robot est amené à traverser une zone contrainte, comme un stent ou un trocart par exemple, qui peuvent créer des frottements importants parallèlement à l'axe du robot. Dans la suite de cette étude, nous distinguerons donc les cas avec et sans cette hypothèse **(H1)** en désignant par *forces 2D* des forces soumises à **(H1)** et par *forces 3D* celles non soumises à **(H1)**. Ces deux types de forces seront donc étudiés séparément et comparés pour étudier l'impact de l'hypothèse **(H1)** sur l'estimation de force.

Nous avons vu dans la sous-section 3.4.3 que la matrice de compliance généralisée permet de modéliser des déformations générées par plusieurs forces appliquées simultanément. L'objectif de cette étude étant de prouver le concept d'estimation de forces par mesure de position, nous allons dans un premier temps considérer un scénario simple

pour lequel la déformation mesurée est générée par une seule force. Cette hypothèse sera désignée par la suite par **(H2)**. Il faut toutefois garder à l'esprit que la méthode présentée par la suite a été conçue et implémentée de manière à gérer des scénarios plus complexes faisant intervenir simultanément plusieurs forces. Avec l'hypothèse **(H2)**, la formulation analytique (4.11) reste valable, bien qu'il n'y ait alors plus qu'un seul terme dans la somme, et par conséquent la formulation matricielle, initialement (4.12), devient :

$$\forall s_0 \in [0, L_0], \begin{bmatrix} \Delta p_0(0, s_0) \\ \vdots \\ \Delta p_0(s, s_0) \\ \vdots \\ \Delta p_0(L_0, s_0) \end{bmatrix} \approx \begin{bmatrix} C_{s_0}^*(0) \\ \vdots \\ C_{s_0}^*(s) \\ \vdots \\ C_{s_0}^*(L_0) \end{bmatrix} \Delta f_0(s_0) \quad (4.14)$$

que l'on désignera de manière plus compacte par :

$$\forall s_0 \in [0, L_0], \Delta \mathcal{P}_0(s_0) \approx C_{s_0}^* \Delta f_0(s_0) \quad (4.15)$$

et le problème d'optimisation associé devient :

$$\Delta \tilde{f}_0(\tilde{s}_0) = \underset{\Delta f_0(s_0)}{\operatorname{argmin}} \left(\|\Delta \mathcal{P}_0 - \Delta \mathcal{P}_0(s_0)\|^2 \right) \quad (4.16a)$$

$$= \underset{\Delta f_0(s_0)}{\operatorname{argmin}} \left(\|\Delta \mathcal{P}_0 - C_{s_0}^* \Delta f_0(s_0)\|^2 \right) \quad (4.16b)$$

Dans l'équation (4.16a), $\Delta \mathcal{P}_0(s_0)$ désigne la déformation estimée du robot tandis que $\Delta \mathcal{P}_0$ désigne la déformation réelle et mesurée. Dans le cas d'une estimation de force idéale, $\Delta \mathcal{P}_0(s_0)$ et $\Delta \mathcal{P}_0$ seraient identiques. Il convient de préciser que la norme $\|\bullet\|$ peut être appliquée à la variable $\Delta \mathcal{P}_0 - C_{s_0}^* \cdot \Delta f_0(s_0)$ car elle est par construction de taille $3N_P \times 1$.

Lors de l'optimisation (4.16b), l'objectif est de trouver 4 variables différentes : les trois composantes du vecteur $\Delta \tilde{f}_0$ ainsi que l'abscisse du point d'application \tilde{s}_0 . Nous proposons alors de scinder le problème d'optimisation (4.16b) en deux : un problème d'optimisation dédié à l'estimation du vecteur de variation de forces $\Delta \tilde{f}_0$ (4.17a) et un second dédié à l'estimation de l'abscisse du point de d'interaction \tilde{s}_0 (4.17b) :

$$\forall s_0 \in [0, L_0], \Delta \tilde{f}_0(s_0) = \underset{\Delta f_0}{\operatorname{argmin}} \left(\|\Delta \mathcal{P}_0 - C_{s_0}^* \Delta f_0\|^2 \right) \quad (4.17a)$$

$$\tilde{s}_0 = \underset{s_0}{\operatorname{argmin}} \left(\|\Delta \mathcal{P}_0 - C_{s_0}^* \Delta \tilde{f}_0(s_0)\|^2 \right) \quad (4.17b)$$

L'optimisation (4.17a) permet de retrouver, pour un point de contact s_0 donné, le vecteur force $\Delta \tilde{f}_0(s_0)$ qui a généré la déformation linéarisée la plus proche de celle mesurée $\Delta \mathcal{P}_0$. Après avoir calculé (4.17a) pour tous les $s_0 \in [0, L_0]$, l'optimisation (4.17b) reprend les estimations des vecteurs forces $\Delta \tilde{f}_0(s_0)$ obtenues et les compare pour sélectionner celle qui a généré la déformation la plus réaliste. En sachant que l'optimisation (4.17a) est quadratique et que la matrice de compliance généralisée est calculée au préalable pour

toutes les valeurs $(s, s_0) \in [0, L_0]^2$, les optimisations (4.17a) peuvent donc être parallélisées pour réduire le temps de calcul total nécessaire à l'estimation de la force $\Delta \tilde{f}_0(\tilde{s}_0)$. De plus, si le nombre de points de discrétisation reste raisonnable, l'optimisation (4.17b) peut être réalisée en effectuant une recherche exhaustive parmi les résultats obtenus afin d'éviter d'éventuels minima locaux.

4.2.5 Estimation de l'incertitude

A cause du mauvais conditionnement du problème, on peut s'attendre à observer, pour des déformations très similaires, des estimations de forces très différentes. En admettant que l'erreur de mesure de la déformation puisse être représentée par une variable aléatoire de variance $\text{Var}(\Delta \mathcal{P}_0)$ connue, nous proposons de quantifier la variance $\text{Var}(\Delta \tilde{f}_0(\tilde{s}_0))$ induite. Pour arriver à cet objectif, il faut tout d'abord expliciter la valeur de $\Delta \tilde{f}_0(s_0)$, calculée par (4.17a), pour pouvoir en extraire la variance. Pour cela, nous introduisons la fonction $G(\Delta f_0)$ définie comme la fonction de coût impliquée dans (4.17a) :

$$G(\Delta f_0) = \|\Delta \mathcal{P}_0 - \mathcal{C}_{s_0}^* \Delta f_0\|^2 \quad (4.18)$$

qui peut être développée comme suit :

$$G(\Delta f_0) = \Delta \mathcal{P}_0^T \Delta \mathcal{P}_0 - 2\Delta f_0^T \mathcal{C}_{s_0}^{*T} \Delta \mathcal{P}_0 + \Delta f_0^T \mathcal{C}_{s_0}^{*T} \mathcal{C}_{s_0}^* \Delta f_0 \quad (4.19)$$

La fonction $G(\Delta f_0)$ est, par définition, positive et convexe. Son minimum global est donc atteint lorsque son gradient s'annule :

$$\left(\frac{\partial G}{\partial \Delta f_0}(\Delta f_0) = 0 \right) \implies \left(\mathcal{C}_{s_0}^{*T} \Delta \mathcal{P}_0 = \mathcal{C}_{s_0}^{*T} \mathcal{C}_{s_0}^* \Delta f_0 \right) \quad (4.20)$$

Puisque l'estimation $\Delta \tilde{f}_0$ est le vecteur qui minimise la fonction $G(\Delta f_0)$, l'équation (4.20) nous donne donc une expression explicite pour $\Delta \tilde{f}_0$:

$$\forall s_0 \in [0, L_0], \Delta \tilde{f}_0(s_0) = \left(\mathcal{C}_{s_0}^{*T} \mathcal{C}_{s_0}^* \right)^{-1} \mathcal{C}_{s_0}^{*T} \Delta \mathcal{P}_0 \quad (4.21)$$

Finalement, on obtient la variance :

$$\forall s_0 \in [0, L_0], \text{Var}(\Delta \tilde{f}_0(\tilde{s}_0)) = A \text{Var}(\Delta \mathcal{P}_0) A^T \quad (4.22)$$

où

$$A = \left(\mathcal{C}_{s_0}^{*T} \mathcal{C}_{s_0}^* \right)^{-1} \mathcal{C}_{s_0}^{*T} \quad (4.23)$$

Grâce à l'équation (4.22), il devient donc possible de propager l'incertitude de la mesure $\Delta \mathcal{P}_0$ sur l'estimation $\Delta \tilde{f}_0(\tilde{s}_0)$. La variance $\text{Var}(\Delta \tilde{f}_0(\tilde{s}_0))$, de taille (3×3) , peut alors être représentée sous la forme d'une ellipse probabiliste. La [figure 4.3](#) illustre comment cette ellipse peut permettre de représenter un cône de force couvrant l'estimation de la force pour différents niveaux de confiance. Cette indication graphique est naturellement dépendante d'un facteur d'échelle arbitraire nécessaire pour représenter graphiquement les intensités de force.

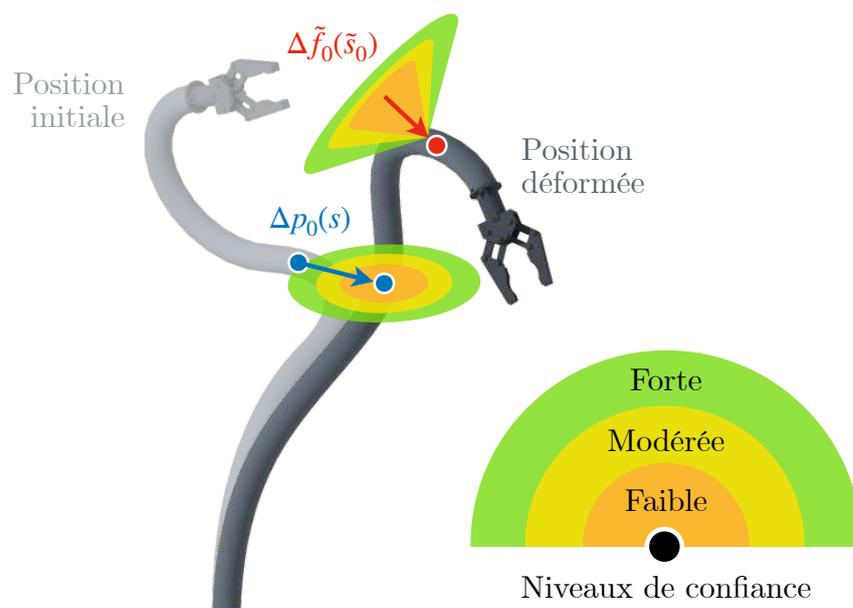


FIGURE 4.3 – Illustration de la variance de l'estimation de la force $\Delta \tilde{f}_0(\tilde{s}_0)$ qui peut être calculée sous condition de pouvoir quantifier la variance de toutes les déformations $\Delta p_0(s)$.

4.3 Cas d'étude : Estimation des forces pour les robots à tubes concentriques

Nous avons présenté dans la [section 4.2](#) une nouvelle formulation du problème d'estimation de forces à partir d'une mesure de forme. Cette nouvelle approche, rendue possible grâce au concept de matrice de compliance généralisée, est appliquée dans cette section aux RTC. Une première étude, réalisée en simulation, présente dans la [sous-section 4.3.1](#) les performances de l'estimation de forces. Avec la volonté de tester expérimentalement cette méthode, nous présentons à la [sous-section 4.3.2](#) la réalisation d'un banc d'essai expérimental, spécialement développé pour cet objectif, comprenant notamment un prototype de RTC. Enfin, les résultats expérimentaux obtenus sont présentés et discutés à la [sous-section 4.3.3](#).

4.3.1 Validation numérique

L'estimation de force a été testée en simulation sur la configuration [RTC3-C1](#) dont la forme initiale est notée $p^*(s)$. Une force 3D $\|\Delta f_0(s_0 = 145)\| = 8.0$ N a ensuite été appliquée sur $p^*(s)$ et la forme déformée, notée $p_{mes}(s)$ a été calculée en résolvant à nouveau le modèle de forme direct. La déformation du robot, notée $\Delta p(s)$ est alors mesurée comme la différence entre $p^*(s)$ et $p_{mes}(s)$. La matrice de compliance généralisée $C_{s_0}(s)$ associée à $p^*(s)$ ainsi que la déformation $\Delta p(s)$ sont alors utilisées pour résoudre le problème d'optimisation (4.16). L'estimation de $\Delta \tilde{f}_0(\tilde{s}_0)$ a été effectuée deux fois, avec et sans l'hypothèse (H1), sachant que la force simulée $\Delta f_0(s_0)$, c'est-à-dire la vérité terrain, ne suit pas l'hypothèse (H1). Une fois l'estimation de la force effectuée, deux nouvelles formes sont calculées :

- La forme déformée du robot, notée $p_{mod}(s)$, calculée en résolvant à nouveau le modèle de forme direct en incluant la force estimée :

$$p_{mod}(s) = \mathcal{M}(s, \Omega_{rob}, \Omega_{crlt} + \Delta \tilde{f}_0(\tilde{s}_0)) \quad (4.24)$$

- La forme déformée du robot, notée $p_{jacob}(s)$, calculée à partir d'une linéarisation de la déformation :

$$p_{jacob}(s) = p^*(s) + C_{\tilde{s}_0}^*(s) \Delta \tilde{f}_0(\tilde{s}_0) \quad (4.25)$$

L'estimation 3D de la force $\Delta \tilde{f}_0(\tilde{s}_0)$, ainsi que les différentes formes, sont illustrées par la [figure 4.4](#). On observe que l'estimation 3D $\Delta \tilde{f}_0(\tilde{s}_0)$ est proche de la force simulée $\Delta f_0(s_0)$: l'erreur sur l'amplitude est de 0.1 N et l'erreur sur la localisation du point de contact est de 5 mm, c'est-à-dire un pas de discrétisation. Lors de la [sous-section 4.2.2](#), nous avons montré qu'il y a une corrélation entre l'estimation de l'amplitude $\|\Delta \tilde{f}_0(\tilde{s}_0)\|$ et le point d'application \tilde{s}_0 . Dans les cas de l'estimation de force et pour une même déformation mesurée, plus le point de contact estimé \tilde{s}_0 est proche de la base, plus l'amplitude estimée $\|\Delta \tilde{f}_0(\tilde{s}_0)\|$ sera élevée.

Puisque l'optimisation (4.17a) est réalisée pour tous les points d'application de force possibles, c'est-à-dire $\forall s_0 \in [0, L_0]$, on peut alors visualiser l'évolution de la fonction de coût (4.17a) en fonction de s_0 , comme présentée à la figure 4.5.

On observe que la valeur de la fonction de coût atteint son minimum global pour \tilde{s}_0 et que les estimations 2D et 3D partagent toutes les deux un même minimum global atteint en $\tilde{s}_0 = 140$ mm. Toutefois, on peut observer que la valeur de la fonction de coût pour $\tilde{s}_0 = 140$ mm n'est pas nulle : l'algorithme utilisé pour résoudre (4.17a) n'a donc pas réussi à trouver la valeur exacte de la force qui a généré la déformation observée $\Delta \mathcal{P}_0$.

Afin de confronter les estimations de force 2D et 3D obtenues par (4.16b), nous proposons de comparer $\Delta \tilde{f}_0(\tilde{s}_0)$ et $\Delta f_0(s_0)$ grâce à trois critères : l'erreur relative sur l'estimation du point d'application $\frac{|s_0 - \tilde{s}_0|}{L_0}$, l'erreur relative sur l'amplitude de la force $\frac{\|\Delta f_0(s_0) - \Delta \tilde{f}_0(\tilde{s}_0)\|}{L_0}$ et l'erreur sur la direction de la force $\alpha(\Delta f_0(s_0), \Delta \tilde{f}_0(\tilde{s}_0))$. Les résultats de la simulation présentée à la figure 4.4 sont exposés dans le tableau 4.1.

L'erreur pour l'estimation du point d'application et pour l'amplitude sont respectivement de 3% et 2% pour les estimations 2D et 3D. L'erreur sur la direction de la force est quant à elle plus faible pour le cas 3D (4% d'erreur) que pour le cas 2D (8%). Cette différence était attendue car dans le cas 3D, la direction de la force estimée $\Delta \tilde{f}_0(\tilde{s}_0)$ n'est pas contrainte par l'hypothèse (H1) : la direction de la force peut donc sortir du plan transversal et être mieux estimée par l'algorithme d'estimation de force.

Pour généraliser les performances de ce nouvel algorithme d'estimation de force, nous avons décidé de le tester à grande échelle sur des forces choisies aléatoirement dans une plage de valeurs réalistes par rapport à la raideur et la longueur du robot. A partir des observations réalisées lors des premiers tests, nous avons sélectionné les plages de valeurs (4.26).

$$1N \leq \|\Delta f_0(s_0)\| \leq 8N \quad (4.26a)$$

$$10\% \leq \frac{s_0}{L_0} \leq 90\% \quad (4.26b)$$

Ces deux plages de valeurs ont été discrétisées en 40 valeurs uniformément réparties, formant ainsi des couples de valeurs $(\Delta f_0(s_0), s_0)$. Tous ces couples de valeurs ont été testés individuellement pour 10 directions aléatoires de force. L'estimation 2D étant contrainte dans le plan transversal, on s'attend donc à observer systématiquement des erreurs $\alpha(\Delta f_0(s_0), \Delta \tilde{f}_0(\tilde{s}_0))$ plus faibles pour le cas 3D que pour le cas 2D. Pour éviter de biaiser les mesures des performances de l'algorithme, nous avons décidé de ne simuler que des forces 2D pour les estimations 2D et que des forces 3D pour les estimations 3D. Au total, pour chaque type d'estimation 2D et 3D, 16000 forces différentes ont été simulées sur la configuration RTC3-C1 et les résultats obtenus ont permis d'extraire les performances statistiques globales de l'estimation de force. Les résultats obtenus sont présentés par le tableau 4.2.

On observe que l'erreur médiane sur l'estimation du point d'application de la force est de 5% et que l'erreur médiane sur l'estimation de l'amplitude de la force est de 17%-18% pour les cas 2D et 3D. Une différence entre les deux types d'estimation apparaît sur l'erreur médiane relative à la direction de la force, passant de 1% pour le cas 2D à 32% pour le cas 3D. Cette différence indique qu'il semble plus difficile d'estimer la composante tangentielle de la force appliquée sur le robot que les deux autres composantes transversales. On peut alors penser que la direction tangentielle du robot est, d'une certaine manière, liée au mauvais conditionnement de la matrice de compliance généralisée $C_{s_0}(s)$. Cette observation sera approfondie dans la [sous-section 4.4.3](#).

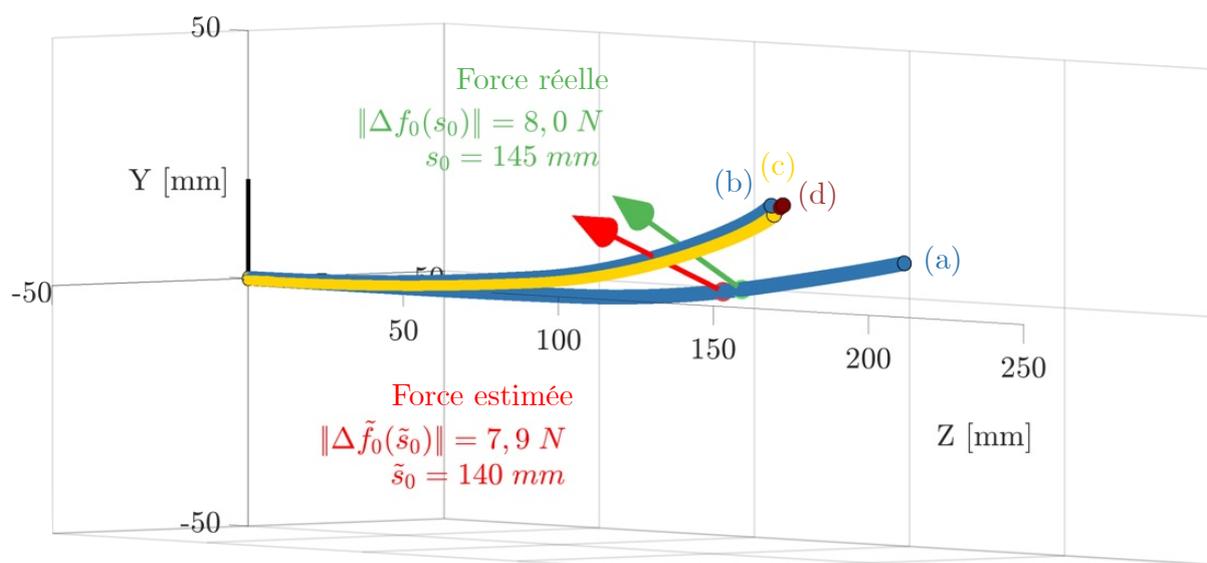


FIGURE 4.4 – Résultats obtenus grâce à l'algorithme d'estimation de force en simulation pour la configuration [RTC3-C1](#). On y retrouve la force simulée $\Delta f_0(s_0)$, la force estimée $\Delta \tilde{f}_0(\tilde{s}_0)$, ainsi que : (a) la forme initiale $p^*(s)$, (b) une simulation de la forme déformée et mesurée $p_{mes}(s)$, (c) la forme déformée $p_{mod}(s)$ estimée par le modèle du RTC et (d) la forme déformée $p_{jacob}(s)$ estimée par linéarisation grâce à la matrice compliance généralisée.

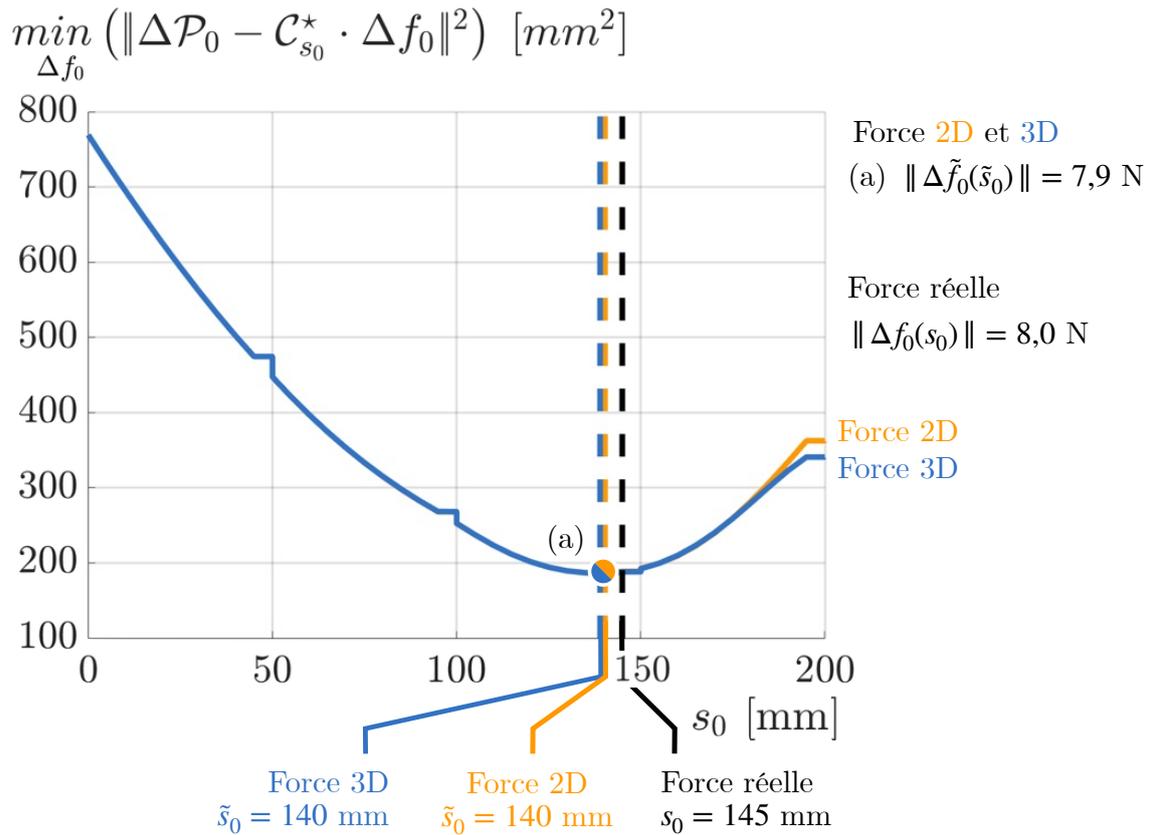


FIGURE 4.5 – Évolution de la fonction de coût simulée du problème d'optimisation (4.17a) en fonction du potentiel point d'application de la force s_0 pour la configuration RTC3-C1. Les minimums des fonctions de coût associées aux estimations 2D et 3D (a) sont atteints pour $\tilde{s}_0 = 140 \text{ mm}$.

Grandeur	Unité	Valeur	
Force simulée			Type de force estimée
s_0	mm	145	
$\ \Delta f_0(s_0)\ $	N	8.0	
Force estimée			
\tilde{s}_0	mm	140	2D
		140	3D
$\ \Delta \tilde{f}_0(\tilde{s}_0)\ $	N	7.9	2D
		7.9	3D
$\frac{ s_0 - \tilde{s}_0 }{L_0}$	%	3	2D
		3	3D
$\frac{\ \ \Delta f_0(s_0)\ - \ \Delta \tilde{f}_0(\tilde{s}_0)\ \ }{\ \Delta f_0(s_0)\ }$	%	2	2D
		2	3D
$\alpha(\Delta f_0(s_0), \Delta \tilde{f}_0(\tilde{s}_0))$	deg	8	2D
		4	3D

TABLE 4.1 – Résultats des estimations 2D et 3D de forces obtenues en simulation pour la configuration [RTC3-C1](#).

4.3. Cas d'étude : Estimation des forces pour les RTC

Grandeur	Unité	Mesure	Valeur	
Force simulée				Type de force estimée
$\frac{s_0}{L_0}$	%		$10 \leq \bullet \leq 90$	
$\ \Delta f_0(s_0)\ $	N		$1 \leq \bullet \leq 8$	
Force estimée				
$\frac{ s_0 - \tilde{s}_0 }{L_0}$	%	min	0	2D
		med	5	
		IQR	3	
		max	10	
	%	min	0	3D
		med	5	
		IQR	3	
		max	50	
$\frac{\ \Delta f_0(s_0)\ - \ \Delta \tilde{f}_0(\tilde{s}_0)\ }{\ \Delta f_0(s_0)\ }$	%	min	≈ 0	2D
		med	18	
		IQR	23	
		max	108	
	%	min	≈ 0	3D
		med	17	
		IQR	28	
		max	108	
$\alpha(\Delta f_0(s_0), \Delta \tilde{f}_0(\tilde{s}_0))$	deg	min	≈ 0	2D
		med	1	
		IQR	1	
		max	19	
	deg	min	≈ 0	3D
		med	32	
		IQR	29	
		max	105	

TABLE 4.2 – Résultats statistiques des estimations de forces 2D et 3D obtenues en simulation pour les 16000 forces simulées sur la configuration [RTC3-C1](#). *min* indique la valeur minimale, *med* la valeur médiane, *IQR* la valeur de l'écart inter-quartiles et *max* la valeur maximale.

4.3.2 Développement d'un prototype

Suite aux résultats prometteurs observés en simulation et présentés dans la [sous-section 4.3.1](#), nous avons souhaité tester l'algorithme d'estimation d'effort expérimentalement sur un prototype de RTC. Puisque nous ne disposons pas de prototype, il était nécessaire d'en fabriquer un spécialement pour cette occasion. Nous avons donc proposé un stage ingénieur de 6 mois, puis recruté et encadré sur ce travail Chloé Lindingre, étudiante de l'INSA de Strasbourg et de l'École de Technologie Supérieure de Montréal, pour son projet de fin d'études.

Le prototype conçu comprend un mécanisme d'actionnement, illustré [figure 4.6](#), pouvant actionner jusqu'à 4 tubes. En pratique, nous nous sommes limités à l'actionnement de 3 tubes, chacun fixé à une plateforme. Chacune d'elle, actionnée en translation par rapport au bâti, comprend un mandrin actionné en rotation sur lequel le tube est fixé. L'architecture mécanique a été pensée de manière à déporter, à une extrémité du bâti, les actionneurs : 6 moteurs Maxon DCX 22 S sur lesquels des réducteurs planétaires GPX 22 ont été installés ainsi que des encodeurs ENX 10 EASY. Les transmissions des mouvements mécaniques jusqu'aux plateformes et aux mandrins sont assurées respectivement par des vis sans fin pour la translation et une liaison glissière puis un engrenage pour la rotation. Le guidage en translation des plateformes est assuré par des patins glissants IGUS T TWE-04 montés sur des rails IGUS T TS-04 installés sur la partie inférieure du bâti. Les 3 tubes en nitinol utilisés pour le robot ont été achetés à Euroflex GmbH et leur caractéristiques géométriques sont renseignées dans le [tableau 4.3](#).

	Tube 1	Tube 2	Tube 3
Rayon extérieur [μm]	978	762	590
Rayon intérieur [μm]	884	648	445
Épaisseur [μm]	94	114	146
Jeu inter-tubes [μm]	122		58

TABLE 4.3 – Caractéristiques géométriques des sections des tubes du RTC.

D'après le fabricant, le nitinol utilisé, le NiTi SE508, a un module d'Young annoncé compris entre 28 et 41 GPa. Un essai expérimental en flexion réalisé sur les tubes reçus a permis d'estimer le module d'Young réel à environ 65 GPa. Ces tubes, achetés droits, ont été mis en forme grâce à l'aide de Kanty Rabenorosoa et Pierre Roux de l'équipe AS2M du laboratoire Femto-ST de Besançon. Les tubes ont été maintenus en position dans des matrices en acier, disponibles au sein de l'équipe AS2M, et ont ensuite subi un trempage thermique. Ce procédé permet de modifier plastiquement la forme des tubes qui prennent alors la forme de la rainure usinée dans la matrice. Les paramètres de la forme des tubes (L_{ri} , L_{ci} et R_{ci}) réellement obtenus ont ensuite été estimés à partir de photographies et grâce à un algorithme de détection de forme. Les paramètres obtenus sont renseignés en annexe dans la configuration [RTC4-C1](#). Finalement, ces 3 tubes ont été montés sur le mécanisme d'actionnement pour former le RTC.

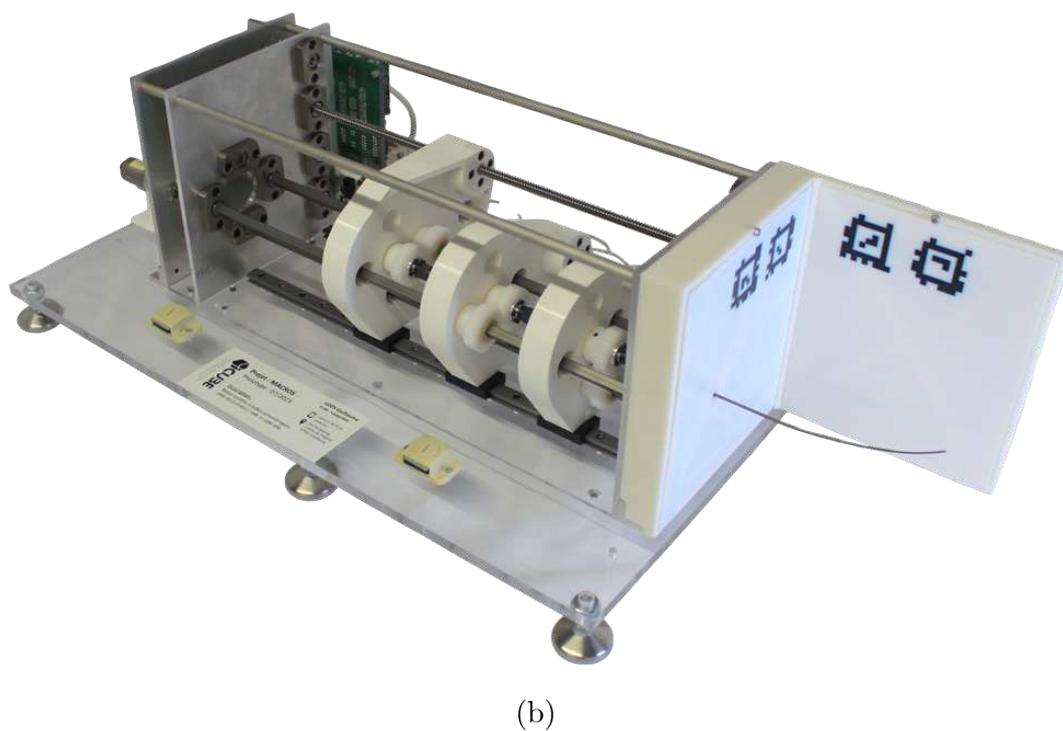
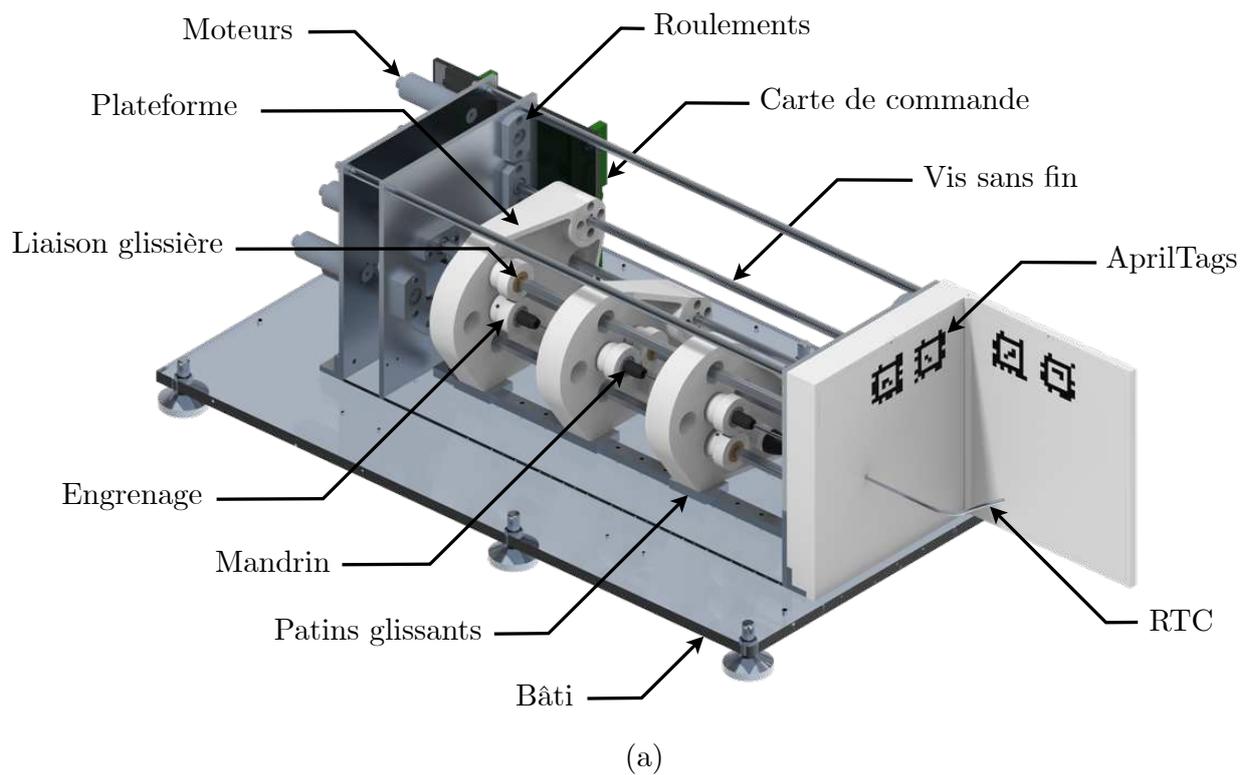


FIGURE 4.6 – Image générée de la conception du prototype de RTC (a) et photographie du prototype réellement fabriqué (b).

Le prototype comprend également un circuit électrique permettant de contrôler les moteurs. Une armoire électrique permet de centraliser une alimentation électrique, divers composants électriques ainsi qu'un ordinateur embarqué Beckhoff CX5140, programmé en C++ par l'ingénieur de recherche Philippe Zanne, membre de l'équipe RDH du laboratoire ICube, à Strasbourg. Ce boîtier électrique dispose d'une façade avant permettant de mettre en marche ou d'arrêter le système électrique, d'informer l'utilisateur de son état courant grâce à des voyants lumineux ainsi que de prévenir des risques grâce à des pictogrammes adaptés. En cas de problème, l'utilisateur peut activer indépendamment deux boutons d'arrêt d'urgence : l'un disposé sur la façade avant et l'autre déporté à un autre endroit accessible du banc de test. Le rôle de l'ordinateur embarqué est de récupérer les consignes des actionneurs, provenant de l'ordinateur principal, et d'assurer le bon fonctionnement de la carte Technosoft iPOS360x MBX6-CAT, fixée à l'arrière du mécanisme, qui commande les moteurs.

Le prototype de RTC a été installé au centre d'un banc expérimental, présenté par la [figure 4.7](#), comprenant un dynamomètre utilisé pour exercer une force d'amplitude connue sur le RTC et fixé sur un bras articulé. Afin de mesurer la forme 3D du robot, nécessaire pour estimer sa déformation, nous avons installé un second bras articulé sur lequel ont été fixées deux caméras : une Prosilica GC 660 et une Mako G-131. Les positions de ces caméras sont estimées grâce à quatre AprilTags visibles à l'avant du mécanisme d'actionnement [[Olson, 2011](#)]. Ces deux caméras permettent d'acquérir deux images du robot sous deux angles de vue différents puis sur chaque image, le RTC sombre devant un fond clair, est détecté par un algorithme de traitement d'images. Un exemple de détection est présenté par la [figure 4.8](#). Enfin, un algorithme de reconstruction épipolaire permet d'estimer la forme 3D du robot à partir des deux détections obtenues.

L'algorithme de reconstruction, illustré par la [figure 4.9](#) et inspiré de [[Franco and Boyer, 2005](#)], consiste à retrouver des points dans l'espace appartenant au RTC. Pour cela, nous commençons par définir un volume initial, discrétisé en voxels, dans lequel se trouve le RTC. Ensuite, chaque voxel est projeté sur les plans images associés aux deux caméras et si la projection se trouve suffisamment proche de la détection du robot, le voxel est alors sélectionné. Il est possible de paralléliser les projections et les sélections des voxels pour couvrir l'intégralité du volume initial. Les voxels sélectionnés forment alors une estimation 3D grossière du RTC. Pour affiner la reconstruction de la forme, il est alors possible de réduire itérativement la taille des voxels et de répéter le processus décrit jusqu'à obtenir une estimation suffisamment fine de la forme 3D du robot. Naturellement, plus la discrétisation est fine, plus le nombre de voxels à traiter est important et donc plus la reconstruction 3D est chronophage. Cette approche itérative, impliquant une taille variable de voxel est un bon compromis entre la rapidité et la précision de l'estimation de la forme.

4.3. Cas d'étude : Estimation des forces pour les RTC

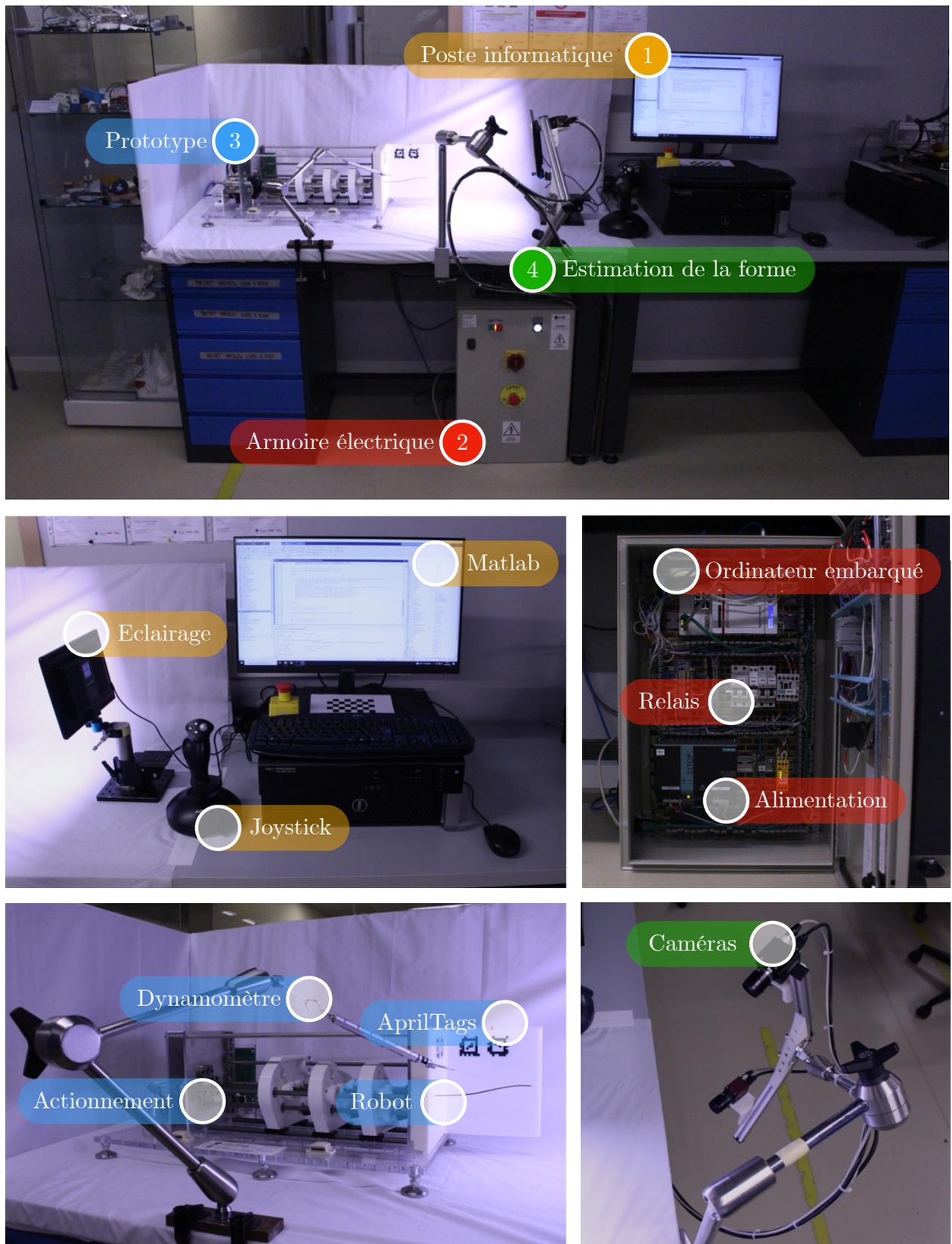


FIGURE 4.7 – Photographies du banc de test comprenant un poste informatique, une armoire électrique, le prototype de RTC ainsi qu'un système d'estimation de la forme du robot.

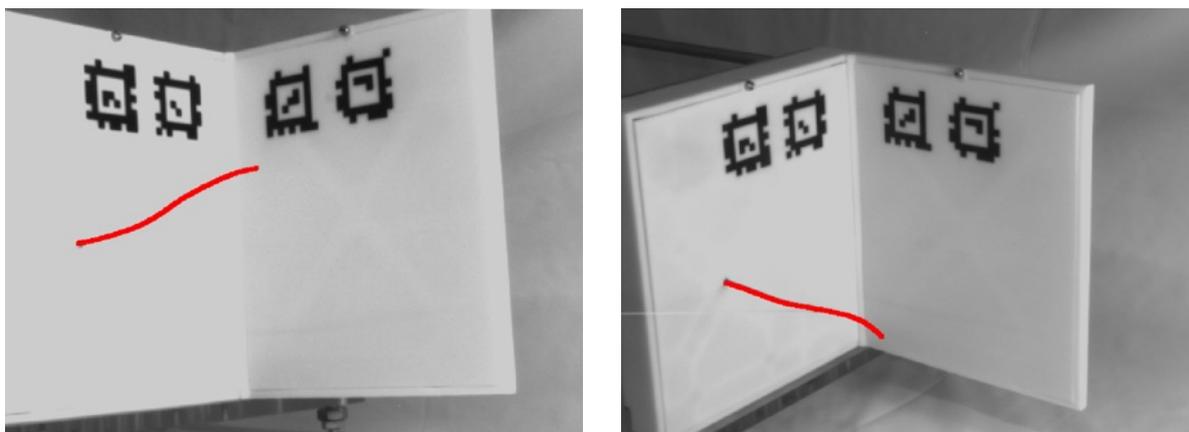


FIGURE 4.8 – Images du RTC, dans la configuration [RTC4-C1](#), prises par les deux caméras. La détection du robot est affichée en rouge sur les images.

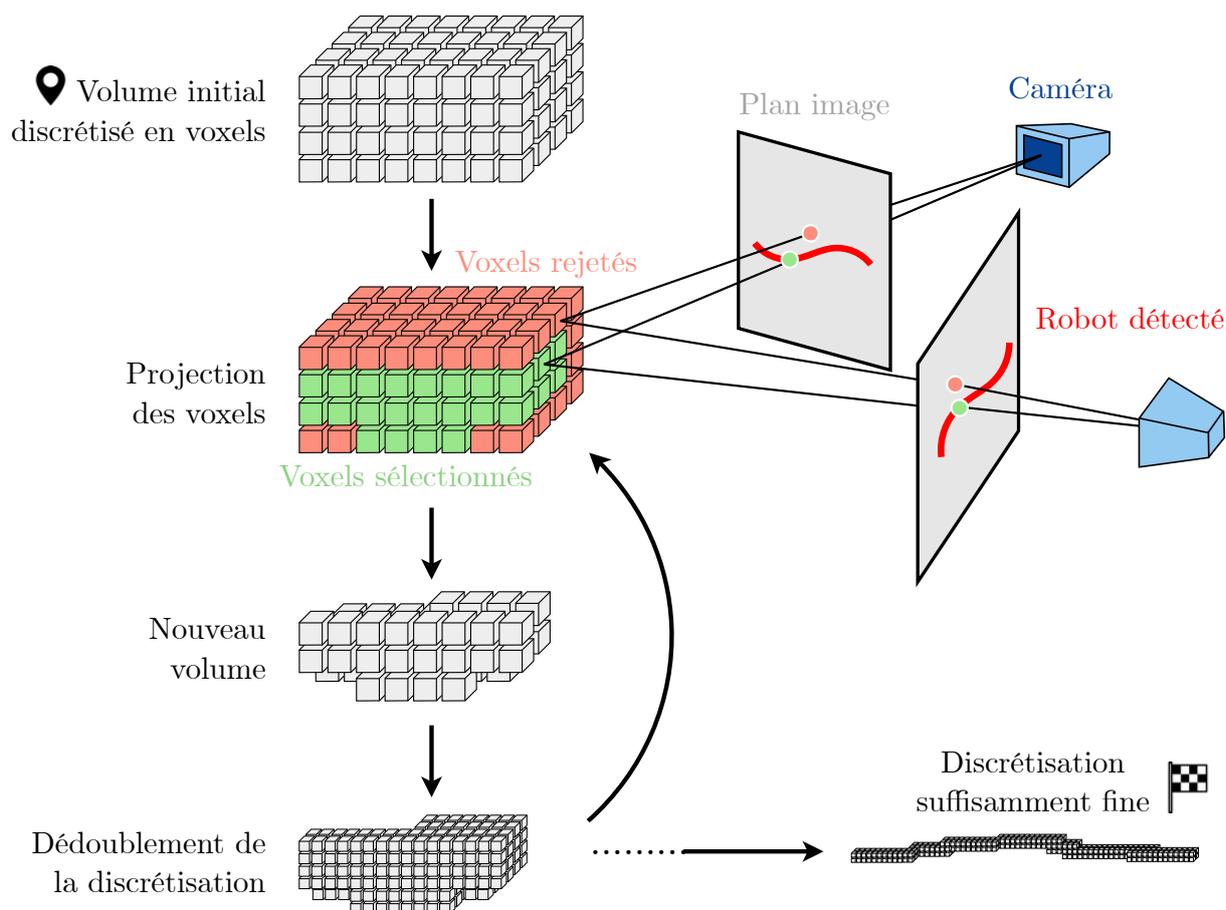


FIGURE 4.9 – Illustration de l'algorithme de reconstruction épipolaire de la forme 3D du RTC, inspiré de [\[Franco and Boyer, 2005\]](#), qui a été implémenté pour le banc de test expérimental.

4.3.3 Résultats expérimentaux

Le banc de test, présenté dans la sous-section 4.3.3, a donc permis de tester expérimentalement notre estimation de force. Nous avons donc déformé notre robot, initialement dans la configuration RTC4-C1, grâce au dynamomètre, puis mesuré la déformation générée en utilisant les deux caméras. La figure 4.10 présente la déformation du robot et la figure 4.11 présente les estimations de forme et de force obtenues.

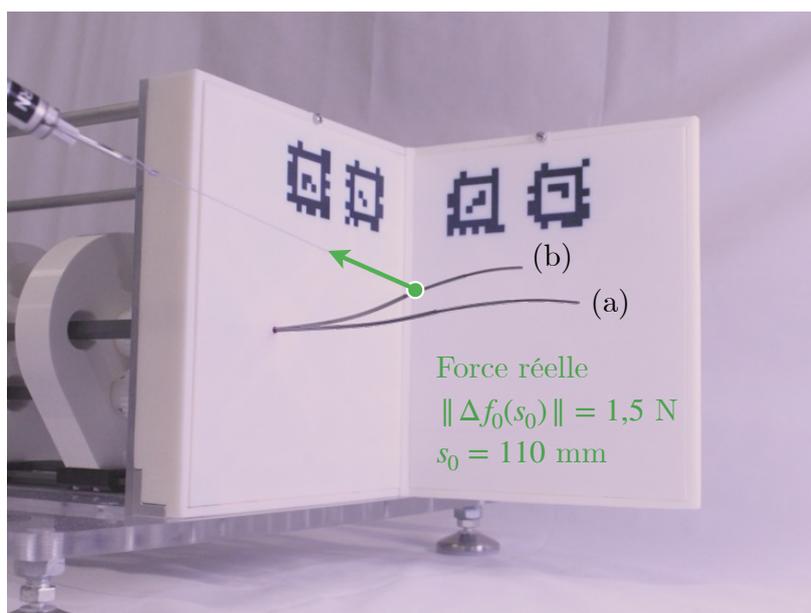


FIGURE 4.10 – Photographies superposées du RTC, dans la configuration RTC4-C1, à l'état initial (a) et lorsqu'il est déformé par une force $\|\Delta f_0(s_0 = 110)\| = 1.5$ N.

On peut observer une erreur entre la forme initiale calculée par le modèle de forme direct (g) et celles estimées par les caméras (e)-(f). Bien que cette erreur soit relativement faible, elle impacte tout de même le réalisme de la matrice de compliance généralisée étant donnée que la forme initiale modélisée ne coïncide pas tout à fait avec la forme initiale observée. Par rapport aux simulations, les résultats expérimentaux sont beaucoup moins satisfaisants : l'erreur sur l'amplitude de la force et sur la localisation du point de contact sont respectivement de 2 N et de 51 mm. Le banc de test utilisé ne permet pas en l'état d'estimer la direction réelle de la force appliquée sur le robot. Pour cette raison, l'erreur $\alpha(\Delta f_0(s_0), \Delta \tilde{f}_0(\tilde{s}_0))$, utilisée en simulation, ne sera pas quantifiée et les forces réelles $\Delta f_0(s_0)$ appliquées sur le robot seront systématiquement supposées être 3D.

Afin de mieux comprendre l'origine de ces erreurs relativement importantes, il est intéressant d'étudier l'évolution de la fonction de coût (4.17a) en fonction de s_0 , présentée par la figure 4.12. On observe cette fois que les minimums globaux sont éloignés de 51 mm du point d'application nominal et également moins "creusés", en comparaison avec les courbes simulées. Cette observation indique qu'il semble beaucoup plus compliqué d'estimer correctement le point de contact expérimentalement plutôt qu'en simulation. Puisque

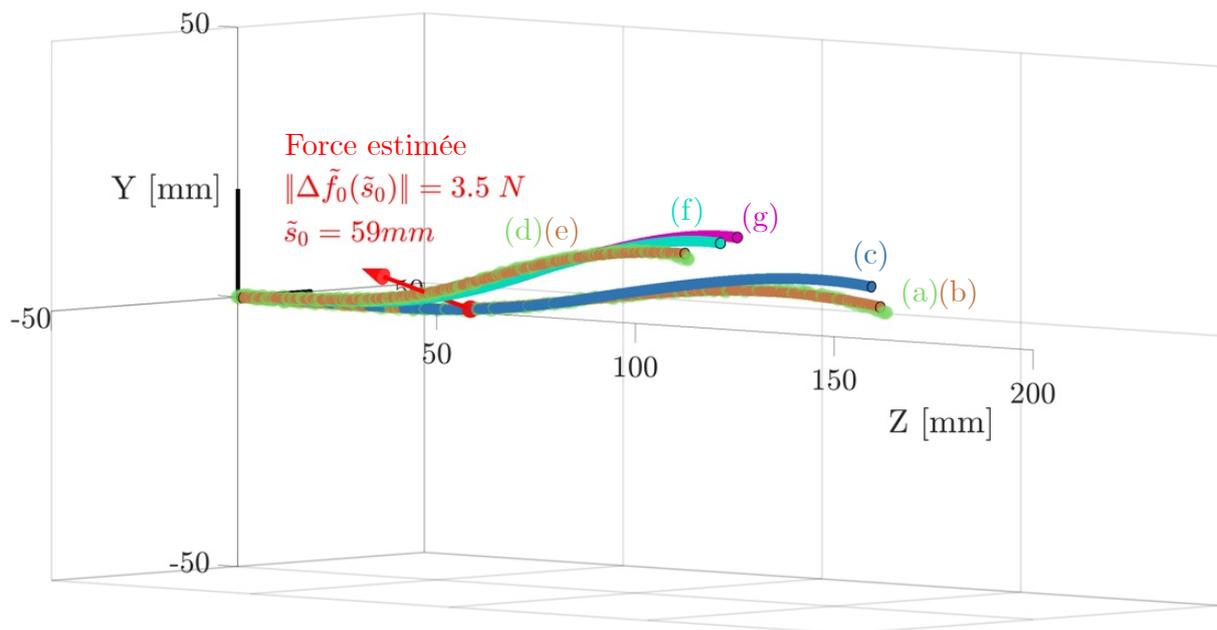


FIGURE 4.11 – Présentation de la force 3D estimée ainsi que des différentes formes estimées : l'estimation brute (a) et sa représentation sous forme de spline (b) de la forme initiale, la forme initiale prédite par le modèle $p^*(s)$ (c), l'estimation brute (d) et sa représentation sous forme de spline (e) de la forme déformée, la forme déformée estimée grâce au modèle $p_{mod}(s)$ (f) et la forme déformée estimée grâce à la compliance généralisée $p_{jacob}(s)$ (g).

que $\Delta\tilde{f}_0$ et \tilde{s}_0 sont intrinsèquement liés, une mauvaise estimation de \tilde{s}_0 va impliquer une mauvaise estimation de $\Delta\tilde{f}_0$. Lorsque l'on injecte l'information du point d'interaction dans l'optimisation (4.17a), on observe que les erreurs sur l'intensité des forces estimées $\Delta\tilde{f}_0(\tilde{s}_0)$ passent respectivement de 267% et 133% pour les cas 2D et 3D à 27% pour les estimations $\Delta\tilde{f}_0(s_0)$ 2D et 3D.

Pour vérifier cette tendance, nous avons réalisé l'estimation de 3 forces appliquées à des points de contact différents : 70 mm, 100 mm et 150 mm sur la configuration RTC4-C1 ayant une longueur déployée totale de $L_0 = 161$ mm. Les résultats obtenus sont regroupés dans le tableau 4.4. On peut tout d'abord observer que les estimations 3D sont presque systématiquement moins bonnes que les estimations 2D, et cela même si la force appliquée ne suit pas l'hypothèse (H1). Cette observation indique que l'estimation en 2D, bien que nécessairement faussée par l'hypothèse (H1), permet tout de même d'améliorer l'estimation de la force.

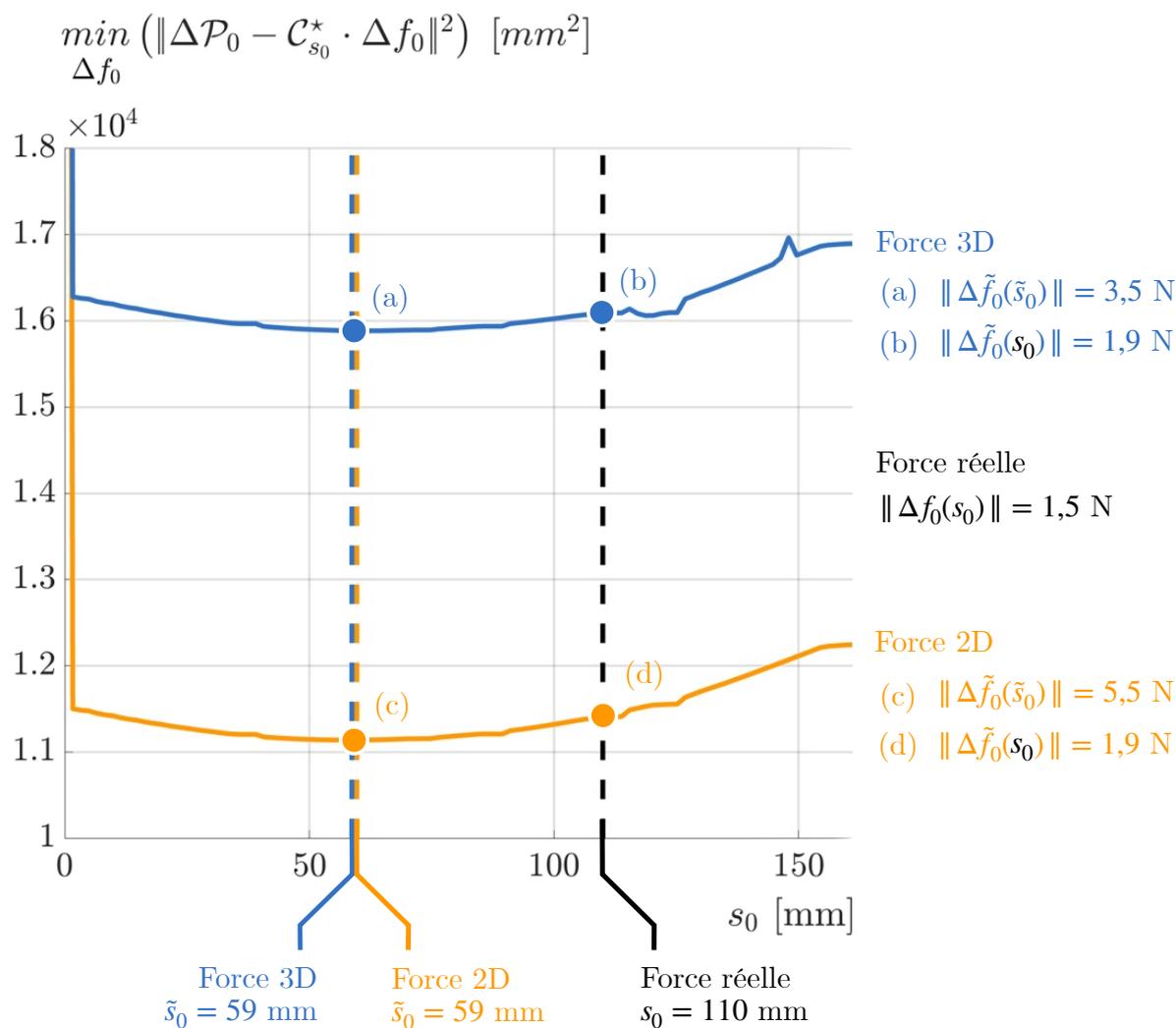


FIGURE 4.12 – Évolution de la fonction de coût du problème d'optimisation (4.17a) en fonction du potentiel point d'application de la force s_0 obtenue lors de l'essai expérimental réalisé sur la configuration RTC4-C1. Les minimums des fonctions de coût associées aux estimations 3D (a) et 2D (c) sont atteints pour $\tilde{s}_0 = 59$. Les estimations 3D (b) et 2D (d) pour les vecteurs forces $\Delta \tilde{f}_0(s_0)$ sont obtenues en injectant l'information du point d'interaction réel s_0 dans le problème d'optimisation (4.17a).

Grandeur	Unité	Essais			
		n°1	n°2	n°3	
Force réelle					Type de force estimée
s_0	mm	70	110	150	
$\ \Delta f_0(s_0)\ $	N	2.9	1.5	1.0	
Force estimée					
\tilde{s}_0	mm	47	59	127	2D
		41	59	133	3D
$\ \Delta \tilde{f}_0(\tilde{s}_0)\ $	N	4.5	3.5	1.1	2D
		5.9	5.5	1.2	3D
$\frac{ s_0 - \tilde{s}_0 }{L_0}$	%	14	32	14	2D
		18	32	11	3D
$\frac{\ \Delta f_0(s_0)\ - \ \Delta \tilde{f}_0(\tilde{s}_0)\ }{\ \Delta f_0(s_0)\ }$	%	55	133	10	2D
		103	267	20	3D
Force estimée connaissant s_0					
$\ \Delta \tilde{f}_0(s_0)\ $	N	2.5	1.3	0.8	2D
		2.5	1.9	1.3	3D
$\frac{\ \Delta f_0(s_0)\ - \ \Delta \tilde{f}_0(s_0)\ }{\ \Delta f_0(s_0)\ }$	%	14	13	20	2D
		14	27	30	3D

TABLE 4.4 – Résultats des trois tests d'estimation de forces obtenus expérimentalement pour la configuration [RTC4-C1](#) pour des points d'interaction situés à 70 mm, 100 mm et 150 mm le long du RTC. Le banc expérimental développé ne permettant pas de mesurer la direction réelle de la force appliquée sur le robot, les erreurs $\alpha(\Delta f_0(s_0), \Delta \tilde{f}_0(\tilde{s}_0))$ n'ont donc pas été quantifiées.

4.4 Discussions et perspectives

Nous avons vu dans ce chapitre comment la matrice de compliance généralisée, présentée dans le [chapitre 3](#), peut être utilisée pour formuler le problème d'estimation d'efforts le long d'un robot continu d'une manière alternative aux méthodes existantes dans l'état de l'art jusqu'alors [[Aloi and Rucker, 2019](#), [Qiao et al., 2021](#), [Aloi et al., 2022](#)]. Après avoir expliqué pourquoi il n'est pas envisageable de résoudre ce problème inverse de déformation par inversion de matrice, nous avons présenté comment construire une méthode de résolution par optimisation sur le problème direct. Cette méthode a montré de bonnes performances en simulation, dans le cas de l'estimation d'une unique force, puis a été évaluée expérimentalement sur un prototype de RTC spécialement conçu pour tester cette approche.

4.4.1 Discussions sur le banc expérimental

Une première remarque que l'on peut faire concerne la qualité des résultats expérimentaux présentés dans la [sous-section 4.3.3](#). On peut observer sur la [figure 4.12](#) un décalage entre la forme initiale mesurée par le système d'acquisition (les formes (a) et (b) sur la figure) et la forme initiale prédite par le modèle du robot $p^*(s)$ (la forme (c) sur la figure). Cette erreur peut s'expliquer par plusieurs facteurs, en particulier une mauvaise identification des paramètres utilisés dans le modèle. Pour y remédier, il pourrait être envisagé de mettre en place une solution de calibration automatique du RTC [[Jang et al., 2016](#)] dédiée à identifier d'éventuels décalages dans les articulations, dans la géométrie des tubes ou même dans les propriétés matérielles des tubes. De façon complémentaire, une analyse de sensibilité visant à quantifier l'erreur induite par ces décalages dans les divers paramètres identifiés pourrait être réalisée, comme dans l'étude menée dans [[Boyer et al., 2024](#)] pour de la commande en boucle fermée. L'erreur observée peut également être due à une imprécision lors de la mesure de la forme du robot à l'aide des caméras. Dans ce cas, la solution d'acquisition de la forme pourrait être améliorée en utilisant des caméras ayant une meilleure résolution, en déplaçant les caméras pour améliorer le point de vue et donc la perception 3D, en augmentant le nombre de caméras pour exploiter des informations trifocales [[Andreff and Tamadazte, 2016](#)] ou, plus radicalement, en utilisant une autre solution de mesure comme un RFB par exemple, similairement aux travaux de [[Qiao et al., 2021](#)]. Les résultats expérimentaux obtenus permettent tout de même de clarifier les défis techniques liés à l'estimation de forces par mesure de forme et ouvrent la voie à de nouvelles perspectives de recherche au niveau méthodologique.

4.4.2 Discussions sur l'étude en simulation

Lors des simulations réalisées sur de nombreuses forces aléatoires, l'objectif était de quantifier les performances globales de l'estimation d'une force. Cependant, nous avons vu que le conditionnement de la matrice de compliance généralisée dépend notamment du point d'application s_0 de la force. Par conséquent, on peut supposer que les performances de l'estimation de force dépendent elles aussi de s_0 . De plus, puisque l'on linéarise les

déformations du robot, ces performances devraient dépendre également des intensités des forces à estimer. La [figure 4.13](#) montre la répartition des erreurs d'estimation sur le point d'application, l'amplitude de la force et sur sa direction. Comme attendu nous observons que les performances de l'estimation varient en fonction de la force à estimer.

On observe notamment qu'il est plus difficile d'estimer l'intensité de la force à mesure que celle-ci diminue (figures (a) et (d)). Concernant l'estimation du point d'application, on constate que l'erreur a tendance à décroître à mesure que la force appliquée s'éloigne de la base et également que l'erreur subit des discontinuités franches pour certaines valeurs de forces (figures (b) et (e)). Une étude complémentaire serait intéressante pour comprendre l'origine de ces discontinuités. Finalement, on remarque que l'erreur sur l'estimation de la direction des forces est maximale dans le cas 2D pour des forces intenses appliquées au bout du robot (figure (c)) tandis qu'elle semble globalement constante pour le cas 3D. Ces figures sont très intéressantes car elles détaillent les performances de la méthode d'estimation de forces et permettent de prendre du recul par rapport aux performances globales annoncées dans la [sous-section 4.3.1](#) et dans la littérature. Ce type d'analyse pourrait également être exploité conjointement avec des algorithmes de régularisation issus de la communauté du traitement de signal [[Benning and Burger, 2018](#)] afin de régulariser le problème inverse d'estimation de force.

4.4.3 Perspectives sur la formulation d'hypothèses

Un autre point important concerne les hypothèses sur la direction des forces estimées. Dans ce chapitre, nous avons examiné deux hypothèses, des forces dites 2D contraintes à s'appliquer perpendiculairement à la tangente locale du robot, et des forces dites 3D non contraintes. L'hypothèse de force 2D **(H1)** est communément utilisée dans la littérature [[Aloi et al., 2022](#), [Qiao et al., 2021](#), [Ferguson et al., 2024](#)]. La [figure 4.14](#) présente un résultat d'estimation de force en simulation pour la configuration [RTC3-C1](#) avec le cône probabiliste calculé grâce à la méthode développée en [sous-section 4.2.5](#) (en supposant que chaque déformation mesurée $\Delta p_0(s)$ suit une variance gaussienne). On observe que le cône de probabilité est en pratique très "aplati", c'est-à-dire que l'estimation est particulièrement incertaine sur la direction portée par le vecteur \vec{e}_\ominus , indiqué dans la [figure 4.14](#). Ce phénomène, également observé pour d'autres simulations, nous informe que la direction portée par \vec{e}_\ominus est en réalité la direction le long de laquelle les forces vont générer le moins de déformation.

Cette observation n'est toutefois pas de nature à valider l'hypothèse de forces 2D **(H1)**. En effet, en décomposant la matrice de compliance généralisée sur ses trois vecteurs singuliers, nous observons qu'une valeur singulière, notée σ_\ominus est systématiquement beaucoup plus petite que les deux autres. Par conséquent, le vecteur \vec{e}_\ominus observé dans la [figure 4.14](#) est en pratique le vecteur singulier associé à la valeur singulière σ_\ominus . Il est d'ailleurs intéressant de noter que le vecteur $\vec{e}_\ominus(s)$ (la généralisation de \vec{e}_\ominus pour des s différents) est parfois éloigné de la tangente $\vec{z}_0(s)$, comme on peut le voir sur la [figure 4.15](#) où $\vec{e}_\ominus(s)$ a été évalué pour toutes les abscisses $s \in [0, L_0]$ et pour une unique abscisse $s_0 = \tilde{s}_0 = 145$ mm.

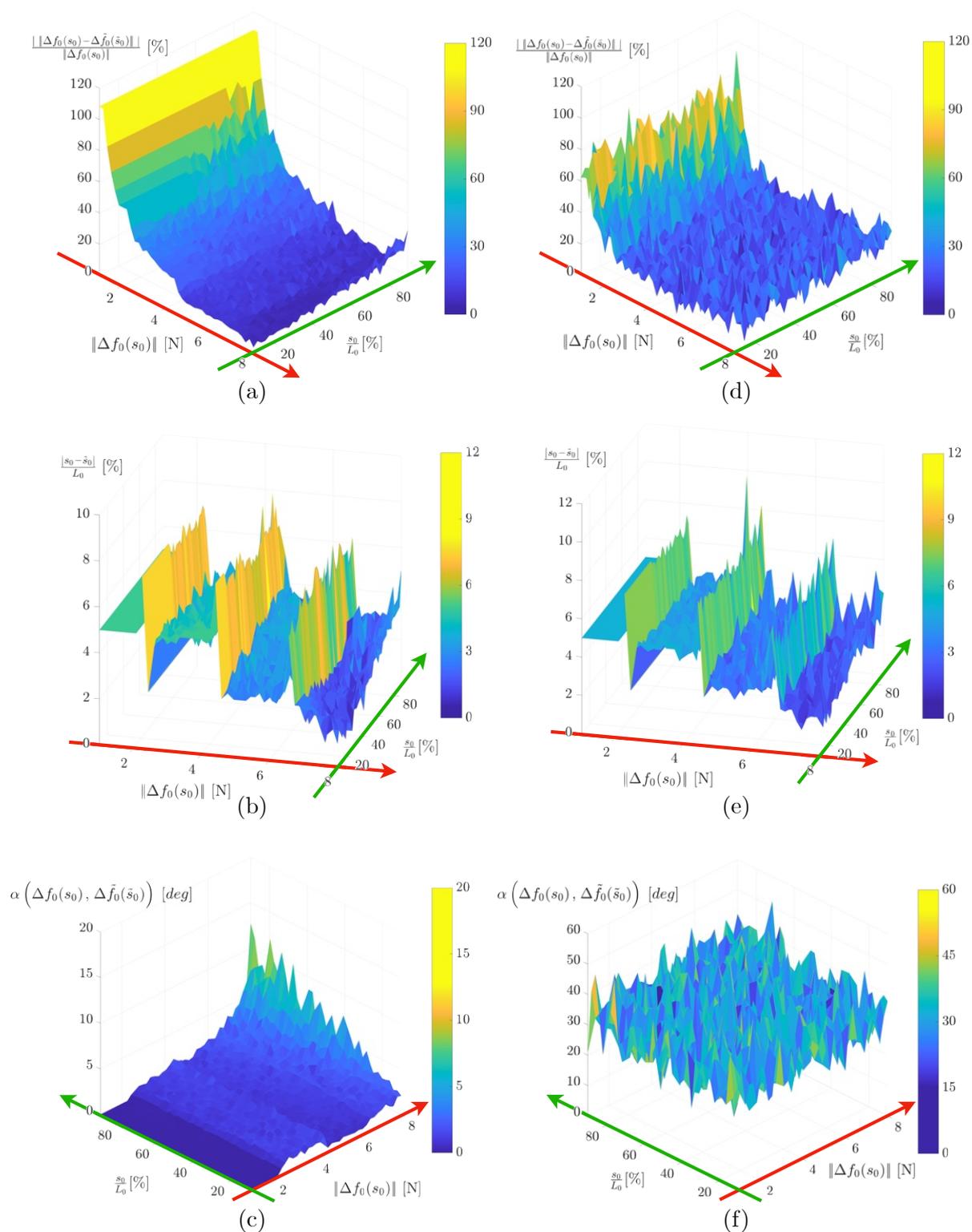


FIGURE 4.13 – Cartographies des erreurs, mesurées lors des essais en simulation de la méthode d'estimation de force, en fonction du point d'application de la force $\frac{s_0}{L_0}$ et de son intensité $\|\Delta f_0(s_0)\|$. On y retrouve les erreurs mesurées pour les estimations 2D (a)-(c) ainsi que pour les estimations 3D (d)-(f). Pour des raisons de visualisation, les orientations des graphiques ne sont pas toutes identiques : les abscisses $\frac{s_0}{L_0}$ croissantes sont indiquées par des flèches vertes et les intensités $\|\Delta f_0(s_0)\|$ croissantes par des flèches rouges.

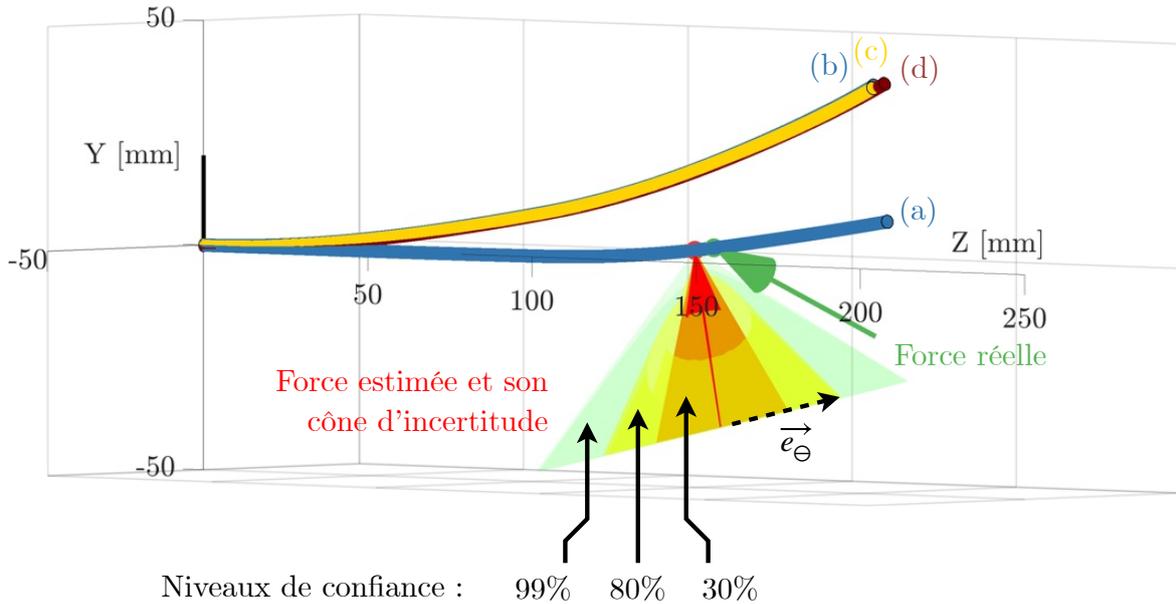


FIGURE 4.14 – Estimation et cône de probabilité de la force appliquée en simulation sur le RTC3. On y retrouve la force simulée $\Delta f_0(s_0)$, la force estimée $\Delta \tilde{f}_0(\tilde{s}_0)$, ainsi que : la forme initiale $p^*(s)$ (a), la forme déformée et mesurée $p_{mes}(s)$ (b), la forme déformée $p_{mod}(s)$ estimée par le modèle du RTC (c) et la forme déformée $p_{jacob}(s)$ estimée par linéarisation grâce à la matrice compliance généralisée (d).

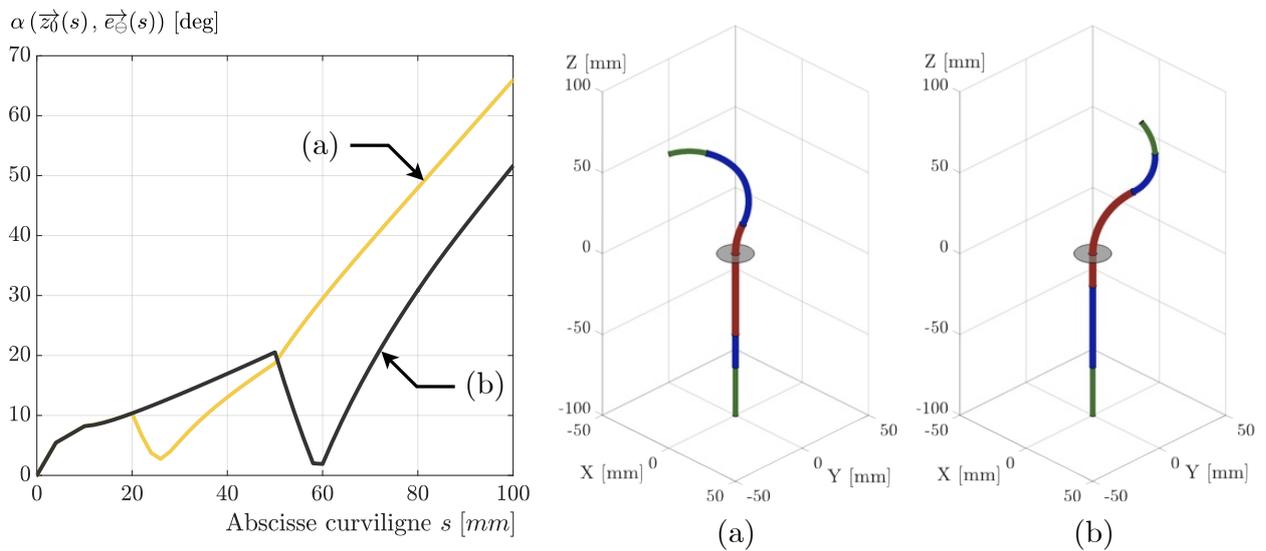


FIGURE 4.15 – Évolution de l'angle entre la tangente du robot $\vec{z}_0(s)$ et la direction de moindre déformation $\vec{e}_\theta(s)$ le long des configurations RTC1-C3 (a) et RTC1-C4 (b).

Sur la [figure 4.15](#), on observe en particulier que les vecteurs $\vec{z}_0(s)$ et $\vec{e}_\ominus(s)$ s'éloignent à mesure que l'on parcourt la longueur du robot et que cela peut varier avec la forme du RTC. Si l'angle $\alpha(\vec{z}_0(s), \vec{e}_\ominus(s))$ restait faible, il aurait été envisageable de passer du repère de base \mathcal{R}_{B_0} au repère flottant $\mathcal{R}_0(s)$ pour résoudre le problème d'estimation d'effort. Cela aurait permis de négliger la composante de force sur $\vec{z}_0(s)$ et de concentrer l'estimation sur les deux composantes de forces associées à $\vec{x}_0(s)$ et $\vec{y}_0(s)$. Cependant, une telle hypothèse ne semble pas adaptée au vu de l'écart entre $\vec{z}_0(s)$ et $\vec{e}_\ominus(s)$. En d'autres termes, une perspective de recherche intéressante consisterait ici à résoudre le problème d'estimation de forces en considérant une hypothèse de forces 2D, non pas dans le repère de base \mathcal{R}_{B_0} ni dans le repère local $\mathcal{R}_0(s)$, comme effectué dans la littérature, mais dans le repère des vecteurs singuliers de $C_{s_0}(s)$, en négligeant la composante de force portée par $\vec{e}_\ominus(s)$. La résolution d'un tel problème d'estimation, soit à partir du problème inverse potentiellement régularisé, soit à l'aide de méthodes d'optimisation, reste toutefois un problème ouvert.

4.4.4 Perspectives sur l'incertitude de l'estimation de force

Même avec une méthode d'estimation de force très performante et une régularisation du problème exploitant les directions singulières de la matrice de compliance généralisée, le calcul de l'incertitude sur les forces estimées resterait certainement indispensable. Lors de la simulation présentée par la [figure 4.14](#), nous avons supposé que chaque incertitude de mesure des points du robot suit une variance gaussienne et donc que ces points ne partagent pas de covariance avec leurs voisins. En pratique, chaque point sur la forme du robot est lié à ses voisins et par conséquent il est probable que les variances $\text{Var}(\Delta p_0(s))$ réelles suivent une distribution plus complexe. Il serait alors intéressant d'implémenter la méthode développée par [[Freundlich et al., 2015](#)] qui permet d'estimer les covariances réelles des points mesurés connaissant les caractéristiques géométriques de la vision stéréo utilisée. Cette nouvelle information permettrait d'améliorer la variance du vecteur force estimé $\text{Var}(\Delta \tilde{f}_0)$. Il serait également intéressant d'étudier l'incertitude sur l'estimation du point d'application des forces en estimant $\text{Var}(\tilde{s}_0)$. Pour pouvoir utiliser la méthode présentée dans la [sous-section 4.2.5](#), il faudrait définir la fonction de coût $G(s_0)$ de manière analogue à (2.39c) puis calculer la quantité $\frac{\partial G}{\partial s_0}(s_0)$, ce qui présente un défi de par le couplage entre $\Delta \tilde{f}_0$ et \tilde{s}_0 présent dans l'équation (4.17b).

Par ailleurs, nous avons vu dans la [sous-section 4.2.5](#) que l'équation (4.21) nous donne une expression explicite du vecteur force optimal $\Delta \tilde{f}_0(s_0)$. Or dans la méthode d'estimation de force présentée, $\Delta \tilde{f}_0(s_0)$ est estimé numériquement grâce à l'optimisation (4.17a) et non avec l'expression explicite (4.21). En pratique l'étude de la covariance a été réalisée après l'implémentation de la méthode par optimisation. Il serait donc intéressant de comparer les résultats issus de l'optimisation numérique avec ceux obtenus par le calcul direct de (4.21).

4.4.5 Perspectives sur le cas multi-forces

Un autre point d'amélioration des méthodes proposées dans ce chapitre concerne le nombre de forces qui s'appliquent sur le robot. Nous avons vu dans le [chapitre 3](#) que la compliance généralisée est adaptée à l'application de plusieurs forces. Cependant lors des simulations et tests expérimentaux réalisés dans le [chapitre 4](#), nous avons considéré que le RTC était déformé par une unique force. Pour illustrer une extension à plusieurs forces, commençons par relever que le problème d'estimation de force posé par les équations (4.17a-4.17b) est interprétable comme la recherche d'un minimum sur une courbe, tel qu'illustré sur la [figure 4.5](#). Une extension à une estimation de deux forces s'appliquant sur le robot serait ainsi la recherche d'un minimum sur une surface et une extension à N_f forces deviendrait alors la recherche d'un minimum dans une hyper-surface inscrite dans un espace de dimension $N_f + 1$. Un avantage de notre formulation est que chaque point de cette hyper-surface peut être calculé indépendamment des autres avec de simples opérations d'algèbre linéaire sur la matrice de compliance généralisée, rendant possible un calcul massivement parallèle. L'interaction entre le nombre de forces N_f et un nombre potentiellement important de minima locaux est cependant difficile à prévoir.

Par ailleurs, dans un cadre général le nombre de forces N_f s'appliquant sur le robot n'est pas connu à l'avance. Dans [[Aloi and Rucker, 2019](#)], ce problème est résolu en résolvant itérativement pour 1 force, puis 2, etc., jusqu'à ce que la valeur de la fonction de coût passe sous un seuil prédéfini. Une autre approche, adoptée dans [[Aloi et al., 2022](#)], consiste à paramétrer les efforts externes selon une distribution dont on recherche les paramètres, limitant ainsi la dimensionnalité du problème de recherche d'efforts externes. La combinaison de ce type de techniques avec la méthode développée dans cette thèse et les observations sur les directions singulières de la matrice de compliance exposées ci-dessus représente une perspective de recherches tout à fait pertinente.

4.4.6 Perspectives générales sur l'estimation d'efforts grâce à une mesure de forme

Pour conclure, en relevant quelques défis techniques, il serait donc possible d'améliorer la méthode d'estimation de force présentée pour la rendre plus performante, applicable à des scénarios impliquant plusieurs forces simultanées et lui permettre de quantifier complètement l'incertitude des forces et des points d'application estimés. Dans la littérature, aucune méthode d'estimation de forces par mesure de la forme du robot n'utilise une linéarisation des déformations. En formulant le problème d'estimation de forces comme un problème d'optimisation quadratique, il devient possible de ré-utiliser les nombreux outils existants dédiés à ce type de problème. De plus, étudier les propriétés de la matrice de compliance généralisée permet d'approfondir la compréhension du problème d'estimation de forces et peut faciliter le développement de nouvelles hypothèses simplificatrices cohérentes.

Chapitre 5

Conclusions et perspectives

5.1 Conclusions générales

Avec l'apparition des robots continus dans les blocs opératoires, de nouvelles procédures minimalement invasives peuvent être utilisées afin de réduire les traumatismes pour les patients bénéficiaires des soins. Ces robots miniatures, totalement ou partiellement commandés à distance par un chirurgien, permettent d'apporter les soins nécessaires dans des zones difficiles d'accès. Pour pouvoir contrôler précisément et de manière fluide la forme de ces robots, il est nécessaire de construire des modèles réalistes et calculables rapidement. Les modèles existants permettent déjà de prédire leur forme avec une erreur limitée et des travaux régulièrement publiés ouvrent la voie à des modèles toujours plus réalistes en se libérant de nombreuses hypothèses.

Avec le développement des robots flexibles, il semblait essentiel d'approfondir la modélisation de la compliance. Les travaux de cette thèse ont permis de construire un modèle de compliance généralisée permettant une description complète de la flexibilité des robots continus. Ce modèle regroupe en un même concept mathématique une information dense pouvant être étudiée et utilisée avec les outils standards de la robotique théorique. Finalement, la compliance généralisée a montré son potentiel applicatif dans le cas de l'estimation d'efforts, qui est encore un verrou scientifique majeur pour une utilisation chirurgicale de robots miniatures et flexibles.

5.2 Perspectives générales

Tout au long de cette thèse, nous avons d'ores et déjà présenté un certain nombre de perspectives à la fin de chaque chapitre, à l'exception du [chapitre 2](#) dont le rôle est de clarifier la modélisation cinétostatique des robots continus. Par conséquent, nous présenterons ci-après des perspectives générales au travail de thèse et à ses contributions.

La contribution centrale de cette thèse, le modèle de compliance généralisée pour les robots continus, présente un certain nombre de limites que nous proposons de discuter. Tout d’abord, ce modèle doit être mis en perspective par rapport aux hypothèses de modélisation explicitées dans le [chapitre 2](#) : élasticité linéaire des matériaux, absence de frottements, et jeu nul entre les tubes pour les RTC. Un certain nombre de travaux ont permis de lever ces hypothèses, afin de proposer des modèles plus réalistes prenant en compte le jeu et la friction entre les tubes d’un RTC [[Ha et al., 2018](#)] ou encore des propriétés non linéaires des matériaux [[Zeng et al., 2021](#), [Shi et al., 2024](#)]. L’incorporation de tels phénomènes –souvent hautement non-linéaires– dans nos modèles représente un problème particulièrement difficile.

Par ailleurs, cette thèse étudie exclusivement les modèles analytiques construits avec une approche newtonienne et rien ne garantit que cette approche soit la plus adaptée. Si, comme relevé dans le [chapitre 3](#), l’extension aux modèles lagrangiens semble une perspective possible à court ou moyen terme, on pourrait également considérer d’autres types de modèles comme les modèles hybrides utilisant localement des modèles expérimentaux, par exemple en couplage avec de la vision [[Sestini et al., 2021](#), [Wang et al., 2024](#)] ou pour des applications de commande [[Liang et al., 2021](#), [Xiang et al., 2023](#)]. Adapter la définition et le calcul de la compliance généralisée sur de tels modèles est une perspective de recherche encore à explorer et qui ouvrirait probablement la voie à de nombreuses nouvelles applications.

Puisque la compliance des robots continus est intrinsèquement liée à leur forme, elle dépend de l’actionnement du robot. Jusqu’à maintenant nous avons caractérisé la compliance sans toutefois envisager de la contrôler à l’aide des actionneurs. En explorant le lien de causalité entre l’actionnement et la compliance, similairement aux travaux de [[Peyron and Burgner-Kahrs, 2023](#)] dans le cas d’une lame actionnée par câbles, il semblerait envisageable de moduler la compliance de l’intégralité du robot à travers son actionnement ou éventuellement en concevant de nouveaux types d’actionneurs dédiés. Ce type d’approche a déjà été étudié dans la littérature, soit par ajout d’éléments mécaniques dans la structure du robot que l’on peut mobiliser pour le rigidifier [[Santiago et al., 2016](#)], soit via la commande en exploitant de l’apprentissage machine [[Jakes et al., 2019](#)]. Une perspective intéressante de nos travaux concerne l’intégration de nos méthodes dans des algorithmes de conception optimale, de commande ou de planification de trajectoire [[Torres et al., 2012](#), [Bergeles et al., 2015](#)] afin de permettre des déploiements chirurgicaux plus sûrs, par exemple en maximisant la souplesse du robot en cas de contact inévitable avec un tissu biologique fragile.

Un autre aspect important concerne le retour d’informations pour le chirurgien. Actuellement, un enjeu en robotique chirurgicale consiste à transmettre au chirurgien de nouveaux types d’informations comme des retours d’efforts haptiques [[Patel et al., 2022](#)] ou des solutions de réalité augmentée [[Qian et al., 2019](#)]. Lorsque la compliance généralisée a été pensée, l’idée était de regrouper en une même grandeur un maximum d’informations. Cependant les solutions technologiques actuelles ne permettent pas de transmettre au chi-

urgien une telle densité d'information. Le développement d'interfaces hommes-machine adaptées à la transmission de la compliance généralisée est une perspective de recherche à fort potentiel, par exemple à l'aide d'interfaces robotiques souples [Zhao et al., 2022], de systèmes de retour haptique répartis [Saudrais et al., 2024] ou encore d'interfaces prenant elles-mêmes la forme d'un robot continu [Frazelle et al., 2020].

Enfin, sur un registre plus mathématique que robotique, nous avons utilisé tout au long de ces travaux des outils mathématiques standards comme le conditionnement de matrices, les ellipsoïdes d'incertitude ou encore la décomposition en valeurs singulières. Ces outils sont très adaptés aux matrices de deux dimensions mais trouvent leur limites lorsque l'on étudie la matrice de compliance généralisée organisée selon quatre dimensions. Une conséquence est que les analyses réalisées dans cette thèse sont systématiquement appliquées sur des "tranches" de compliance, c'est-à-dire pour une valeur (s, s_0) donnée. Le désavantage de ce type d'approche est que la structure de l'information de compliance selon s et s_0 est ainsi perdue. Il convient ici de noter que les fondements théoriques de la robotique ont adapté et développé des méthodes provenant de l'analyse matricielle afin de proposer les outils d'analyse utilisés dans cette thèse, par exemple dans [Yoshikawa, 1985]. De la même manière, il serait intéressant de développer des outils mathématiques fondés sur l'analyse tensorielle [Itskov, 2000] afin d'étudier et exploiter la compliance généralisée dans sa globalité.

Annexes

Caractéristiques					
	Caractéristique	Unité	Valeur		
			Tube $n^{\circ}1$	Tube $n^{\circ}2$	Tube $n^{\circ}3$
RTC1	Longueur rectiligne L_{ri}	mm	20	80	150
	Longueur courbée L_{ci}	mm	50	70	50
	Longueur totale L_i	mm	70	150	200
	Rayon de courbure R_{ci}	mm	30	30	30
	Raideur transversale kx_i	N.mm ²	$1.5 \cdot 10^6$	$4.0 \cdot 10^5$	$2.0 \cdot 10^5$
	Coefficient de Poisson	-	0.3	0.3	0.3
Configurations					
	Actionnement	Unité	Valeur		
			Tube $n^{\circ}1$	Tube $n^{\circ}2$	Tube $n^{\circ}3$
RTC1-C1	Rotation θ_{ci}	deg	115	-65	90
	Déploiement β_{ci}	mm	20	80	100
RTC1-C2	Rotation θ_{ci}	deg	115	-65	180
	Déploiement β_{ci}	mm	20	80	100
RTC1-C3	Rotation θ_{ci}	deg	115	-65	-45
	Déploiement β_{ci}	mm	20	80	100
RTC1-C4	Rotation θ_{ci}	deg	115	-65	-45
	Déploiement β_{ci}	mm	50	80	100
Réglages du solveur					
Paramètres			Unité	Valeur	
Seuil de convergence ϵ			-	10^{-20}	
Pas de discrétisation $\Delta(s)$ standard			mm	2	
Nombre de points avant et après chaque point de discontinuité			-	5	
Pas de discrétisation $\Delta(s)$ près des points de discontinuité			mm	10^{-10}	
Nombre total de points de discrétisation			-	181	
Valeur par défaut pour $y_u(0)$			-	$0_{U \times 1}$	

TABLE 5.1 – Caractéristiques et configurations du RTC1

Caractéristiques					
	Caractéristique	Unité	Valeur		
			Tube n°1	Tube n°2	Tube n°3
RTC2	Longueur rectiligne L_{ri}	mm	15	75	135
	Longueur courbée L_{ci}	mm	30	45	45
	Longueur totale L_i	mm	45	120	180
	Rayon de courbure R_{ci}	mm	150	100	50
	Raideur transversale kx_i	N.mm ²	$1.5 \cdot 10^6$	$4.0 \cdot 10^5$	$2.0 \cdot 10^5$
	Coefficient de Poisson	-	0.3	0.3	0.3

Configurations					
	Actionnement	Unité	Valeur		
			Tube n°1	Tube n°2	Tube n°3
RTC2-C1	Rotation θ_{ci}	deg	0	0	0
	Déploiement β_{ci}	mm	30	90	120

Réglages du solveur			
Paramètres		Unité	Valeur
Seuil de convergence ϵ		-	10^{-20}
Pas de discrétisation $\Delta(s)$ standard		mm	2
Nombre de points avant et après chaque point de discontinuité		-	5
Pas de discrétisation $\Delta(s)$ près des points de discontinuité		mm	10^{-13}
Nombre total de points de discrétisation		-	175
Valeur par défaut pour $y_u(0)$		-	$0_{U \times 1}$

TABLE 5.2 – Caractéristiques et configurations du RTC2

Caractéristiques					
	Caractéristique	Unité	Valeur		
			Tube n°1	Tube n°2	Tube n°3
RTC3	Longueur rectiligne L_{ri}	mm	165	235	450
	Longueur courbée L_{ci}	mm	50	50	50
	Longueur totale L_i	mm	215	285	500
	Rayon de courbure R_{ci}	mm	160	242	118
	Raideur transversale kx_i	N.mm ²	$2.2 \cdot 10^5$	$1.2 \cdot 10^5$	$6.0 \cdot 10^4$
	Coefficient de Poisson	-	0.3	0.3	0.3

Configurations					
	Actionnement	Unité	Valeur		
			Tube n°1	Tube n°2	Tube n°3
RTC3-C1	Rotation θ_{ci}	deg	90	180	270
	Déploiement β_{ci}	mm	100	150	200

Réglages du solveur			
Paramètres		Unité	Valeur
Seuil de convergence ϵ		-	10^{-15}
Pas de discrétisation $\Delta(s)$ standard		mm	5
Nombre de points avant et après chaque point de discontinuité		-	3
Pas de discrétisation $\Delta(s)$ près des points de discontinuité		mm	10^{-8}
Nombre total de points de discrétisation		-	144
Valeur par défaut pour $y_u(0)$		-	$0_{U \times 1}$

TABLE 5.3 – Caractéristiques et configurations du RTC3

Caractéristiques					
	Caractéristique	Unité	Valeur		
			Tube n°1	Tube n°2	Tube n°3
RTC4	Longueur rectiligne L_{ri}	mm	95	215	353
	Longueur courbée L_{ci}	mm	50	50	47
	Longueur totale L_i	mm	145	265	400
	Rayon de courbure R_{ci}	mm	160	242	118
	Raideur transversale kx_i	N.mm ²	$1.6 \cdot 10^4$	$0.8 \cdot 10^3$	$0.42 \cdot 10^3$
	Coefficient de Poisson	-	0.3	0.3	0.3

Configurations					
	Actionnement	Unité	Valeur		
			Tube n°1	Tube n°2	Tube n°3
RTC4-C1	Rotation θ_{ci}	deg	180	90	-90
	Déploiement β_{ci}	mm	90	126	161

Réglages du solveur			
Paramètres		Unité	Valeur
Seuil de convergence ϵ		-	10^{-20}
Pas de discrétisation $\Delta(s)$ standard		mm	10
Nombre de points avant et après chaque point de discontinuité		-	3
Pas de discrétisation $\Delta(s)$ près des points de discontinuité		mm	10^{-8}
Nombre total de points de discrétisation		-	102
Valeur par défaut pour $y_u(0)$		-	$0_{U \times 1}$

TABLE 5.4 – Caractéristiques et configurations du RTC4

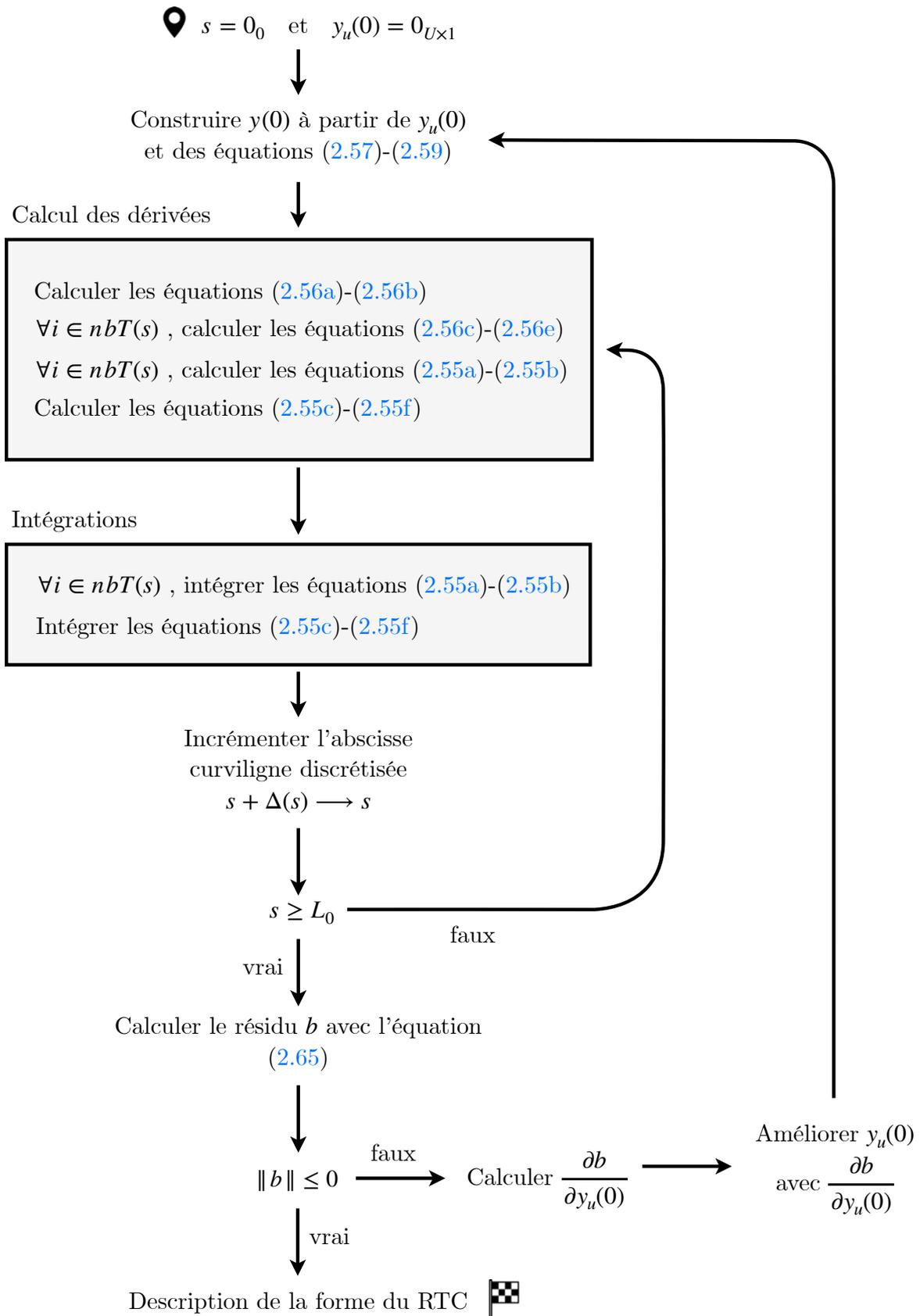


FIGURE 5.1 – Synthèse de la résolution du modèle cinetostatique direct dans le cas d'un RTC

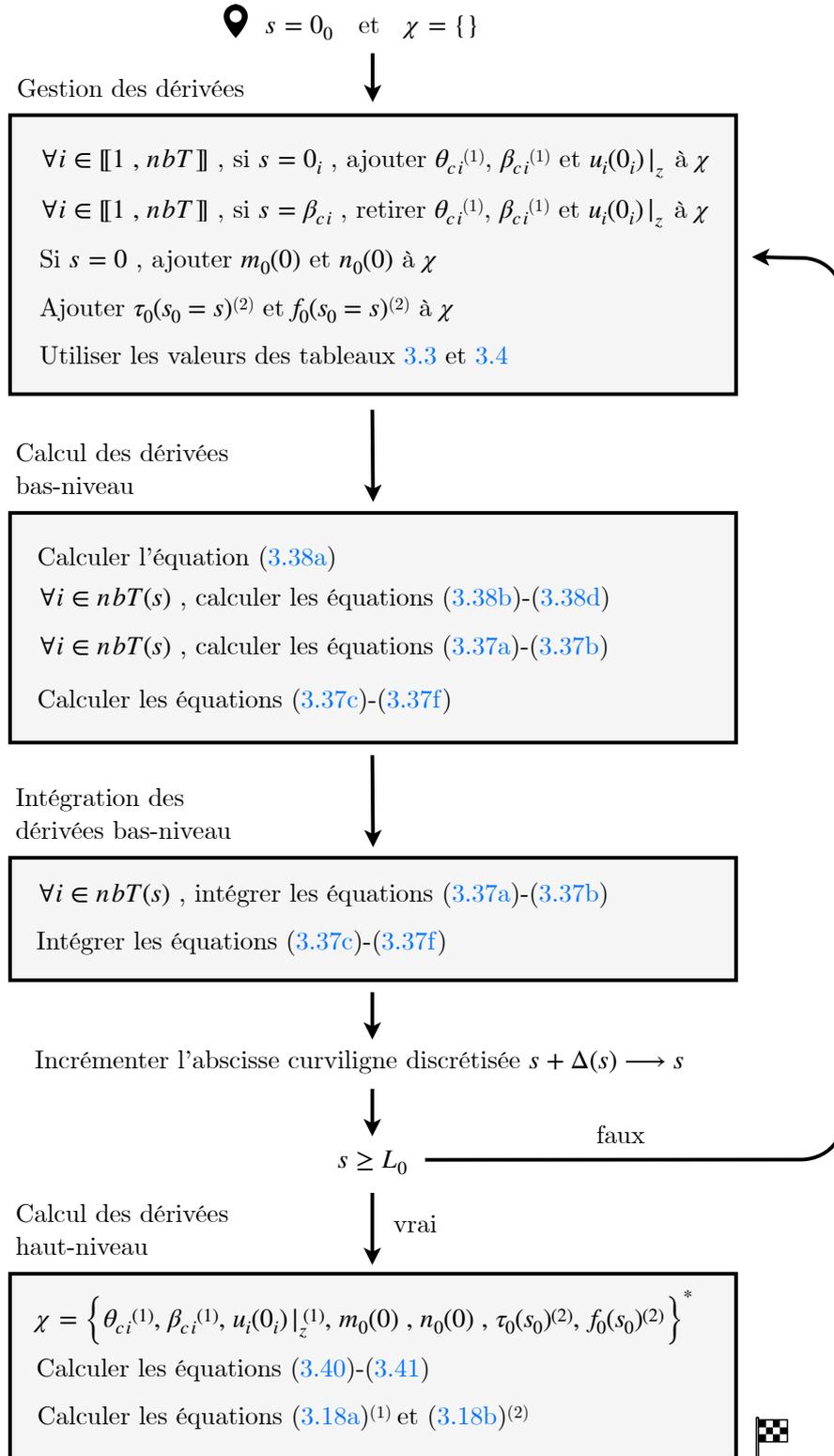


FIGURE 5.2 – Synthèse de la MPDBN dans le cas d'un RTC. L'indication \bullet^* précise ici que le dérivateur χ est considéré pour $\forall i \in \llbracket 1, nbT \rrbracket$ et pour $\forall s_0 \in [0, L_0]$. Les étapes indiquées par $\bullet^{(1)}$ ne sont nécessaires que pour le calcul de la jacobienne articulaire $J(s)$ et $\bullet^{(2)}$ que pour le calcul de la matrice de compliance généralisée $C_{s_0}(s)$.

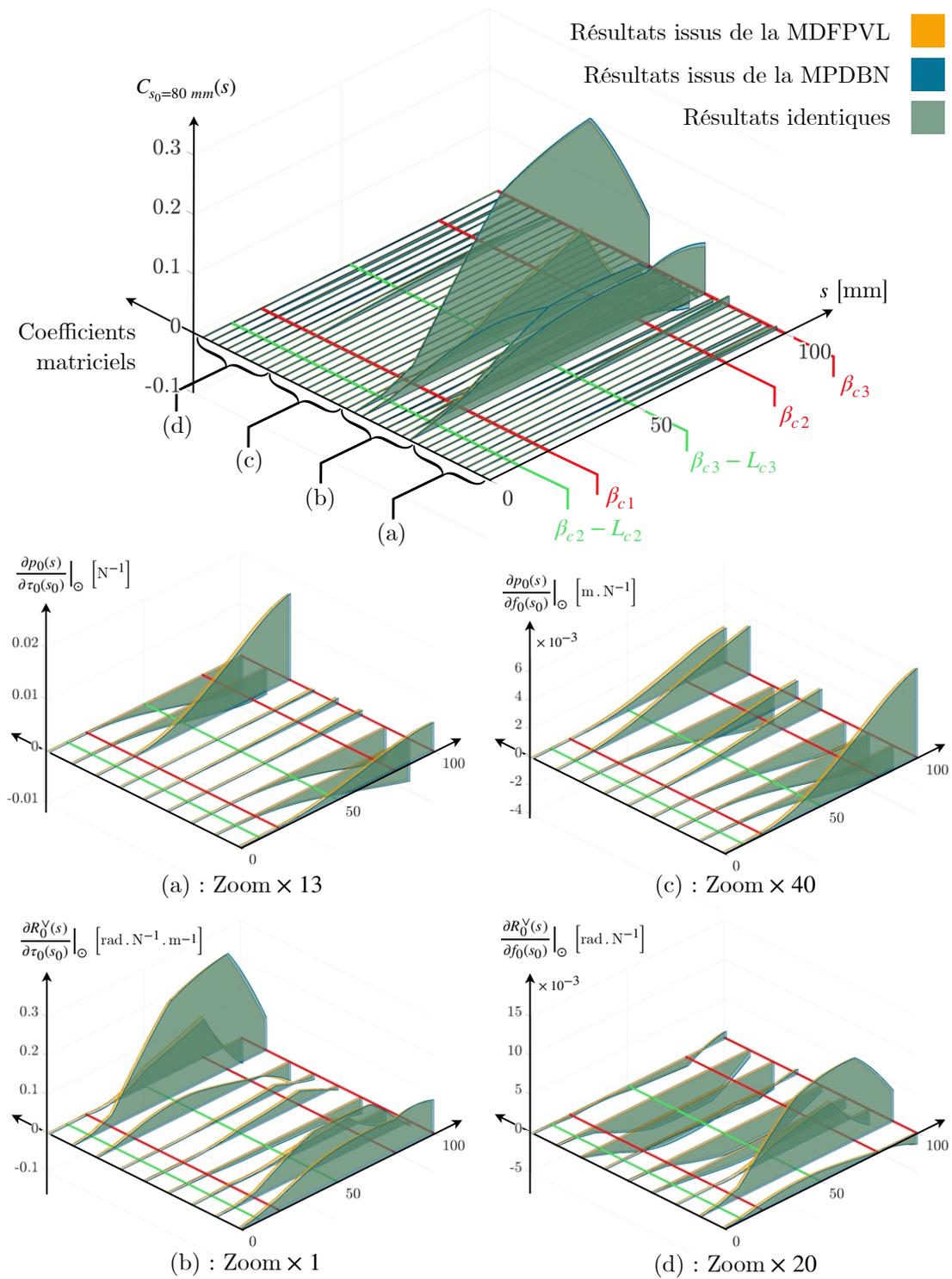


FIGURE 5.3 – Comparaison graphique des coefficients de la matrice de compliance généralisée $C_{s_0=80}(s)$ pour le RTC1-C2. Les courbes de référence, issues de la MDFPVL, sont indiquées en orange et celles obtenues par la MPDBN en bleu.

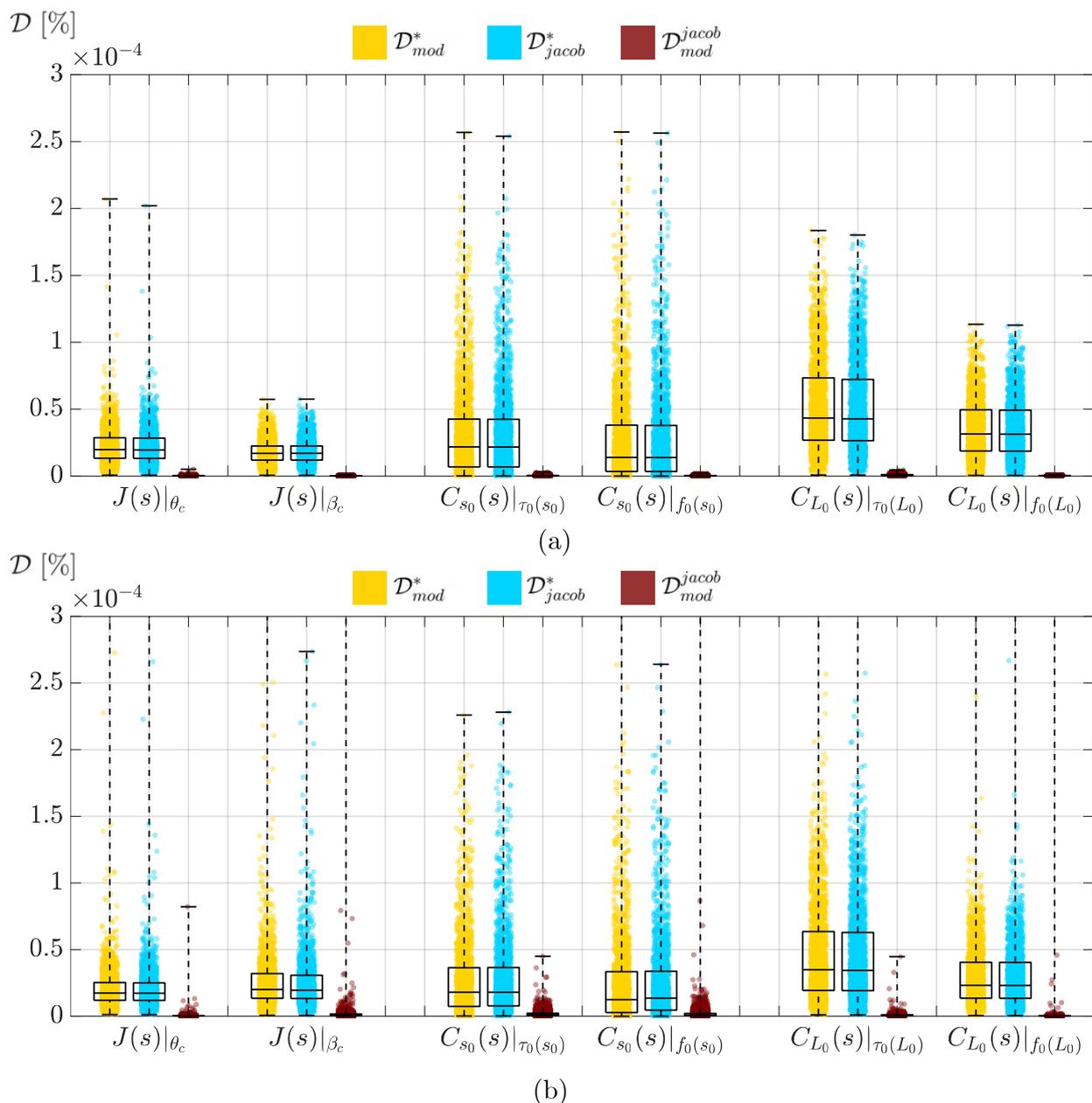


FIGURE 5.4 – Déformations relatives mesurées à partir des formes simulées grâce au modèle (\mathcal{D}_{*}^{mod}) et grâce aux matrices $C_{s_0}(s)$ et $J(s)$ obtenues grâce à la MPDBN (\mathcal{D}_{*}^{jacob}). La différence entre ces deux déformations est mesurée par ($\mathcal{D}_{mod}^{jacob}$). Les résultats présentés ont été obtenus à partir de 5000 simulations de formes sans précharge (a) et avec précharge (b). Contrairement à la validation numérique présentée dans la [sous-section 3.3.3](#) où les variations $\Delta\theta_{ci}$, $\Delta\beta_{ci}$, $\Delta\tau_0(s_0)$, $\Delta f_0(s_0)$, $\Delta\tau_0(L_0)$ et $\Delta f_0(L_0)$ sont appliquées simultanément, les variations sont ici appliquées indépendamment pour trouver des variations limites générant des déformations d'amplitude semblable.

Abréviations	
RTC	Robot à Tubes Concentriques "Concentric Tubes Continuum Robot" (CTCR) en anglais
RC	Robot actionné par Câbles "Tendon-Actuated Continuum Robot" (TACR) en anglais
RM	Robot actionné grâce à un champ Magnétique "Magnetic Continuum Robot" (M-CR) en anglais
RAA	Robot Agoniste-Antagoniste "Concentric Agonist-Antagonist Robot" (CAAR) ou "Concentric Push-Pull Robot" (CPPR) en anglais
RE	Robot à Expansion "Growing Robot" (GR) ou "Vine Robot" (VR) en anglais
PVI	Problème d'intégration aux Valeurs Initiales "Initial Value Problem" (IVP) en anglais
PVL	Problème d'intégration aux Valeurs aux Limites "Boundary Value Problem" (BVP) en anglais
MT	Méthode de résolution par Tirages "Shooting Method" en anglais
MDFPVL	Méthode par Différences Finies sur le Problème d'intégration aux Valeurs Limites
MDFPVI	Méthode par Différences Finies sur le Problème d'intégration aux Valeurs Initiales
MPDHN	Méthode par Propagation de Dérivées Haut-Niveau "Derivative Propagation Method" (DPM) en anglais
MPDBN	Méthode par Propagation de Dérivées Bas-Niveau "Low-Level Derivative Propagation Method" (LLDPM) en anglais
RFB	Réseau de Fibres de Bragg "Fiber Bragg Grating" (FBG) en anglais

TABLE 5.5 – Listes des abréviations

Opérateurs

Notation	Description
$\dot{\bullet}$	Dérivation par rapport à l'abscisse curviligne s
$\hat{\bullet}/\bullet^\wedge$	Opérateur antisymétrique (2.4)
\bullet^\vee	Opérateur inverse de $\hat{\bullet}/\bullet^\wedge$
\bullet^T	Opérateur de transposition
\times	Produit vectoriel
$\mathcal{F}(\)$	Fonction différentielle sur $y(s)$ (2.19)
$\mathcal{M}(\)$	Modèle cinétostatique direct (2.22)
$\alpha(\)$	Mesure de l'angle entre deux vecteurs
$Rot_z(\)$	Matrice de rotation autour de l'axe z (2.33)
$\ \bullet\ $	Norme euclidienne standard
\bullet^\dagger	Opérateur pseudo-inverse de Moore-Penrose
$d(\)$	Distance entre deux courbes 3D (3.47)

TABLE 5.6 – Nomenclature des opérateurs mathématiques

Symboles périphériques	
Notation	Description
\bullet^*	Fait référence à une variable initiale définie avant un déformation
\bullet_i	Fait référence à une variable définie pour le tube i
\bullet_0	Fait référence à une variable définie pour le robot en général
$\bullet _{x/y/z}$	Fait référence à une composante particulière du vecteur \bullet
$\bullet _{\mathcal{R}}$	Précise que la variable \bullet est définie dans le repère \mathcal{R}
$\bullet _{p_0(s)}$	Fait référence aux composantes de la matrice \bullet relatives à la position $p_0(s)$
$\bullet _{\theta_c / \dots / f_0(L_0)}$	Fait référence aux composantes de la matrice \bullet relatives à $\theta_c / \dots / f_0(L_0)$
$\bullet _{\otimes}$	Indique que la variable \bullet ne respecte pas forcément la contrainte $b = 0$
$\bullet _{\odot}$	Indique que la variable \bullet respecte la contrainte $b = 0$
$\bullet _{\text{“}b\text{”}}$	Indique que la dérivée \bullet est réalisée exclusivement sur l’expression analytique de b et non sur la variable b en tant que telle
$\Delta\bullet$	Fait référence à une variation de la variation \bullet
\bullet_{mod}	Indique que la variable \bullet est calculée grâce au modèle \mathcal{M} du robot
\bullet_{jacob}	Indique que la variable \bullet est calculée par linéarisation grâce à une ou plusieurs jacobiniennes
$\tilde{\bullet}$	Indique que la variable \bullet est estimée

TABLE 5.7 – Nomenclature des symboles périphériques aux variables

Variables des robots continus - Modèle cinétostatique 1/2

Notation	Dimension	Unité	Description
L_0	\mathbb{R}	mm	Longueur curviligne du robot déployé
s	\mathbb{R}	mm	Abscisse curviligne du robot comptée par rapport à \mathcal{R}_{B_0}
s_0	\mathbb{R}	mm	Abscisse curviligne désignant un point sur lequel un effort ponctuel est appliqué
$[l_{min}, l_{max}]$	\mathbb{R}^2	mm	Intervalle d'abscisses curvilignes sur lequel un effort distribué est appliqué
N_P	\mathbb{R}	-	Nombre de points de discrétisation des abscisses curvilignes s
$\Delta(s)$	\mathbb{R}	mm	Pas de discrétisation de l'abscisse curviligne variable en fonction de s
\mathcal{R}_{B_0}	-	-	Repère global et fixe, attaché à la base du robot
$\mathcal{R}_0(s)$	-	-	Repère flottant attaché au robot (2.1)
$T_0(s)$	SE(3)	-	Matrice de transformation rigide de \mathcal{R}_{B_0} vers $\mathcal{R}_0(s)$ (2.2)
$R_0(s)$	SO(3)	rad	Matrice de rotation de \mathcal{R}_{B_0} vers $\mathcal{R}_0(s)$
$p_0(s)$	\mathbb{R}^3	mm	Translation de \mathcal{R}_{B_0} vers $\mathcal{R}_0(s)$
$u_0(s)$	\mathbb{R}^3	mm^{-1}	Variation d'orientation, aussi appelée courbure, de $\mathcal{R}_0(s)$ (2.3a)
$v_0(s)$	\mathbb{R}^3	$\text{rad} \cdot \text{mm}^{-1}$	Variation de position de $\mathcal{R}_0(s)$ (2.3b)
$\xi_0(s)$	\mathbb{R}^6	-	Vecteur regroupant les variations de position et d'orientation $\mathcal{R}_0(s)$ (2.7)
$K_0(s)$	$\mathbb{R}^{3 \times 3}$	-	Matrice de raideur de la section du robot située à l'abscisse s

TABLE 5.8 – Nomenclature des variables associées au modèle cinétostatique des robots continus 1/2

Variables des robots continus - Modèle cinétostatique 2/2			
Notation	Dimension	Unité	Description
$m_0(s)$	\mathbb{R}^3	N.mm	Moment interne au robot
$n_0(s)$	\mathbb{R}^3	N	Force interne au robot
$\tau_0(s_0)$	\mathbb{R}^3	N.mm	Moment appliqué sur le robot au point d'abscisse s_0
$f_0(s_0)$	\mathbb{R}^3	N	Force appliquée sur le robot au point d'abscisse s_0
$w_0(s_0)$	\mathbb{R}^6	-	Vecteur regroupant les forces et moments appliqués sur le robot au point d'abscisse s_0 (2.9)
q	\mathbb{R}^6	-	Vecteur regroupant les variables articulaires du robot (2.23)
$y(s)$	\mathbb{R}^Y	-	Vecteur d'état décrivant le comportement mécanique du robot en s
$y_u(s)$	\mathbb{R}^U	-	Sous-vecteur inclus dans $y(s)$ regroupant les conditions initiales inconnues (2.20)
$y_k(s)$	\mathbb{R}^{Y-U}	-	Sous-vecteur inclus dans $y(s)$, complémentaire à $y_u(0)$, regroupant les conditions initiales connues (2.20)
b	\mathbb{R}^B	-	Résidu calculé en $s = L_0$ dans le cas d'un PVL et qui garantit que le robot est à l'équilibre si $b = 0_{B \times 1}$
Ω_{robot}	-	-	Ensemble de variables désignant les caractéristiques constantes du robot
Ω_{ctrl}	-	-	Ensemble de variables désignant les variables impliquées dans la commande du robot (2.21)

TABLE 5.9 – Nomenclature des variables associées au modèle cinétostatique des robots continus 2/2

Variables des robots à tubes concentriques			
Notation	Dimension	Unité	Description
N_T	\mathbb{N}^*	-	Nombre de tubes total du RTC
$N_T(s)$	\mathbb{N}^*	-	Nombre de tubes du RTC impliqués à l'abscisse curviligne s
\mathcal{R}_{B_i}	-	-	Repère fixe, attaché à la base du tube i
$\mathcal{R}_i(s)$	-	-	Repère flottant attaché au tube i (2.63)
$R_i(s)$	$\text{SO}(3)$	rad	Matrice de rotation de \mathcal{R}_{B_i} vers $\mathcal{R}_i(s)$
0_i	\mathbb{R}	mm	Plus petite abscisse du tube i
0_0	\mathbb{R}	mm	Plus petite abscisse du robot
K_i	$\mathbb{R}^{3 \times 3}$	N.mm ²	Matrice de raideur en flexion de la section du tube i , constante sur toute sa longueur (2.38)
kx_i	\mathbb{R}	N.mm ²	Coefficient de K_i associé à l'axe $\vec{x}_i(s)$ et par hypothèse à l'axe $\vec{y}_i(s)$ également
kz_i	\mathbb{R}	N.mm ²	Coefficient de K_i associé à l'axe $\vec{z}_i(s)$ (2.39d)
$\theta_i(s)$	\mathbb{R}	rad	Torsion du tube i à l'abscisse s (2.35)
$\phi_i(s)$	\mathbb{R}	mm ⁻¹	Variable utilisée pour décrire la courbure constante par morceaux du tube i (2.29)
$u_i^*(s)$	\mathbb{R}^3	mm ⁻¹	Courbure initiale du tube i non déformé (2.28)
$u_i(s)$	\mathbb{R}^3	mm ⁻¹	Courbure du tube i déformé
$v_i(s)$	\mathbb{R}^3	rad.mm ⁻¹	Variation de position de $\mathcal{R}_i(s)$
θ_{ci}	\mathbb{R}	rad	Rotation actionnée de la base du tube i (2.31)
β_{ci}	\mathbb{R}	mm	Déploiement actionné du tube i

TABLE 5.10 – Nomenclature des variables associées aux robots à tubes concentriques

Variables des robots continus - Compliance		
Notation	Dimension	Description
$J(s)$	$\mathbb{R}^{6 \times Q}$	Jacobienne articulaire du robot (3.2)
$C(s)$	$\mathbb{R}^{6 \times 6}$	Matrice de compliance au bout du robot (3.4)
$C_{s_0}(s)$	$\mathbb{R}^{6 \times 6}$	Matrice de compliance généralisée (3.5)
χ	\mathbb{R}	Variable muette utilisée pour représenter les variables de dérivation (3.9)
$E_\chi(s)$	$\mathbb{R}^{6 \times 1}$	Dérivée par rapport à χ de la matrice de transformation rigide $T_0^V(s)$ (3.11)
B_χ	$\mathbb{R}^{B \times 1}$	Dérivée par rapport à χ du résidu b (3.13)
V_χ	$\mathbb{R}^{Y \times 1}$	Dérivée par rapport à χ du vecteur d'état $y(s)$ (3.20)
Y	\mathbb{N}^*	Nombre de composantes du vecteur d'état $y(s)$
U	\mathbb{N}^*	Nombre de composantes du sous-vecteur $y_u(0)$
B	\mathbb{N}^*	Nombre de composantes du résidu b
Q	\mathbb{N}^*	Nombre de variables articulaires regroupées dans q

TABLE 5.11 – Nomenclature des variables associées au modèle de compliance des robots continus

Variables de l'estimation d'efforts			
Notation	Dimension	Unité	Description
\mathcal{C}	$\mathbb{R}^{6N_P \times 6N_P}$	-	Grande matrice de compliance construite en assemblant les "tranches" de la compliance généralisée selon 2 dimensions
$\Delta \mathcal{W}_0$	$\mathbb{R}^{6N_P \times 1}$	-	Grande matrice de variation d'effort construite en assemblant verticalement les variations d'efforts $\Delta w_0(s_0)$, $\forall s_0 \in [0, L_0]$
$\Delta \mathcal{T}_0^\vee$	$\mathbb{R}^{6N_P \times 1}$	-	Grande matrice de déformation construite en assemblant verticalement les variations de pose $\Delta T_0^\vee(s)$, $\forall s \in [0, L_0]$
$\mathcal{C}_{s_0}^*$	$\mathbb{R}^{3N_P \times 3}$	mm.N ⁻¹	Grande matrice de compliance construite en assemblant les "tranches" de la compliance $C_{s_0}^*(s)$ selon 2 dimensions
$\Delta \mathcal{P}_0$	$\mathbb{R}^{3N_P \times 1}$	mm	Grande matrice de déformation en position construite en assemblant verticalement les variations de position mesurées $\Delta p_0(s)$, $\forall s \in [0, L_0]$
$\Delta \mathcal{P}_0(s_0)$	$\mathbb{R}^{3N_P \times 1}$	mm	Grande matrice de déformation en position estimée par linéarisation des déformations grâce à la matrice $\mathcal{C}_{s_0}^*$
$\Delta f_0(s_0)$	\mathbb{R}^3	N	Variation de force réelle ou simulée appliquée sur le robot en s_0
\tilde{s}_0	\mathbb{R}	mm	Estimation du point de contact
$\Delta \tilde{f}_0$	\mathbb{R}^3	N	Estimation de la force appliquée
$\Delta \tilde{f}_0(\tilde{s}_0)$	\mathbb{R}^3	N	Estimation de la force appliquée et du point de contact
$\Delta \tilde{f}_0(s_0)$	\mathbb{R}^3	N	Estimation de la force appliquée connaissant le point de contact

TABLE 5.12 – Nomenclature des variables associées à l'estimation d'efforts

Bibliographie

- [Adagolodjo et al., 2021] Adagolodjo, Y., Renda, F., and Duriez, C. (2021). Coupling numerical deformable models in global and reduced coordinates for the simulation of the direct and the inverse kinematics of soft robots. *IEEE Robotics and Automation Letters*, 6(2) :3910–3917. [79](#)
- [Al-Ahmad et al., 2021] Al-Ahmad, O., Ourak, M., Vlekken, J., and Vander Poorten, E. (2021). Fbg-based estimation of external forces along flexible instrument bodies. *Frontiers in Robotics and AI*, 8 :718033. [85](#), [90](#)
- [Aleluia Porto et al., 2019] Aleluia Porto, R., Nageotte, F., Zanne, P., and de Mathelin, M. (2019). Position control of medical cable-driven flexible instruments by combining machine learning and kinematic analysis. In *2019 international conference on robotics and automation (ICRA)*, pages 7913–7919. IEEE. [12](#)
- [Alian et al., 2023] Alian, A., Mylonas, G., and Avery, J. (2023). Soft continuum actuator tip position and contact force prediction, using electrical impedance tomography and recurrent neural networks. In *2023 IEEE International Conference on Soft Robotics (RoboSoft)*, pages 1–6. IEEE. [84](#)
- [Aloi et al., 2018] Aloi, V., Black, C., and Rucker, C. (2018). Stiffness control of parallel continuum robots. In *Dynamic Systems and Control Conference*, volume 51890, page V001T04A012. American Society of Mechanical Engineers. [49](#)
- [Aloi et al., 2022] Aloi, V., Dang, K. T., Barth, E. J., and Rucker, C. (2022). Estimating forces along continuum robots. *IEEE Robotics and Automation Letters*, 7(4) :8877–8884. [14](#), [85](#), [90](#), [109](#), [110](#), [114](#)
- [Aloi and Rucker, 2019] Aloi, V. A. and Rucker, D. C. (2019). Estimating loads along elastic rods. *2019 IEEE International Conference on Robotics and Automation*, pages 2867–2873. [85](#), [109](#), [114](#)
- [Alqumsan et al., 2019] Alqumsan, A. A., Khoo, S., and Norton, M. (2019). Robust control of continuum robots using cosserat rod theory. *Mechanism and Machine Theory*, 131 :48–61. [14](#)
- [Anderson and Horn, 1970] Anderson, V. C. and Horn, R. C. (1970). Tensor arm manipulator. *United State Patent*, (3,497,083). [6](#)
- [Andreff and Tamadazte, 2016] Andreff, N. and Tamadazte, B. (2016). Laser steering using virtual trifocal visual servoing. *The International Journal of Robotics Research*, 35(6) :672–694. [109](#)
- [Antman, 1995] Antman, S. S. (1995). *Problems in nonlinear elasticity*. Springer. [25](#)

- [Arabagi et al., 2015] Arabagi, V., Felfoul, O., Gosline, A. H., Wood, R. J., and Dupont, P. E. (2015). Biocompatible pressure sensing skins for minimally invasive surgical instruments. *IEEE sensors journal*, 16(5) :1294–1303. [14](#)
- [Ataka et al., 2016] Ataka, A., Qi, P., Shiva, A., Shafti, A., Wurdemann, H., Dasgupta, P., and Althoefer, K. (2016). Towards safer obstacle avoidance for continuum-style manipulator in dynamic environments. In *2016 6th IEEE International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, pages 600–605. IEEE. [48](#)
- [Benning and Burger, 2018] Benning, M. and Burger, M. (2018). Modern regularization methods for inverse problems. *Acta numerica*, 27 :1–111. [110](#)
- [Bentley et al., 2022] Bentley, M., Rucker, C., and Kuntz, A. (2022). Interactive-rate supervisory control for arbitrarily-routed multitendon robots via motion planning. *IEEE Access*, 10 :80999–81019. [28](#)
- [Bergeles et al., 2015] Bergeles, C., Gosline, A. H., Vasilyev, N. V., Codd, P. J., Del Nido, P. J., and Dupont, P. E. (2015). Concentric tube robot design and optimization based on task and anatomical constraints. *IEEE Transactions on Robotics*, 31(1) :67–84. [118](#)
- [Blumenschein et al., 2020] Blumenschein, L. H., Coad, M. M., Haggerty, D. A., Okamura, A. M., and Hawkes, E. W. (2020). Design, modeling, control, and application of everting vine robots. *Frontiers in Robotics and AI*, 7 :548266. [6](#)
- [Boehler et al., 2020] Boehler, Q., Gage, D. S., Hofmann, P., Gehring, A., Chautems, C., Spahn, D. R., Biro, P., and Nelson, B. J. (2020). Realiti : A robotic endoscope automated via laryngeal imaging for tracheal intubation. *IEEE Transactions on Medical Robotics and Bionics*, 2(2) :157–164. [14](#)
- [Boyer et al., 2011] Boyer, F., Ali, S., and Porez, M. (2011). Macrocontinuous dynamics for hyperredundant robots : application to kinematic locomotion bioinspired by elongated body animals. *IEEE Transactions on Robotics*, 28(2) :303–317. [10](#)
- [Boyer et al., 2020] Boyer, F., Lebastard, V., Candelier, F., and Renda, F. (2020). Dynamics of continuum and soft robots : A strain parameterization based approach. *IEEE Transactions on Robotics*, 37(3) :847–863. [10](#), [79](#)
- [Boyer et al., 2024] Boyer, Q., Voros, S., Roux, P., Marionnet, F., Rabenoroso, K., and Chikhaoui, M. T. (2024). On high performance control of concentric tube continuum robots through parsimonious calibration. *IEEE Robotics and Automation Letters*. [45](#), [109](#)
- [Brainlab, 2024] Brainlab (2024). Cirq – neurochirurgie fonctionnelle. Consulté le 7 octobre 2024. [IX](#), [5](#)
- [Brown, 2020] Brown, T. B. (2020). Language models are few-shot learners. *arXiv pre-print arXiv :2005.14165*. [12](#)
- [Burgner-Kahrs, 2022] Burgner-Kahrs, J. (2022). Open continuum robotics project blog. Consulté le 7 octobre 2024. [1](#)
- [Burgner-Kahrs et al., 2015] Burgner-Kahrs, J., Rucker, D. C., and Choset, H. (2015). Continuum robots for medical applications : A survey. *IEEE Transactions on Robotics*, 31(6) :1261–1280. [1](#), [7](#)

- [Camarillo et al., 2008] Camarillo, D. B., Milne, C. F., Carlson, C. R., Zinn, M. R., and Salisbury, J. K. (2008). Mechanics modeling of tendon-driven continuum manipulators. *IEEE transactions on robotics*, 24(6) :1262–1273. [13](#)
- [Cheah et al., 2004] Cheah, C.-C., Liu, C., and Slotine, J.-J. E. (2004). Approximate jacobian adaptive control for robot manipulators. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, volume 3, pages 3075–3080. IEEE. [48](#)
- [Chen et al., 2021a] Chen, Y., Wang, L., Galloway, K., Godage, I., Simaan, N., and Barth, E. (2021a). Modal-based kinematics and contact detection of soft robots. *Soft Robotics*, 8(3) :298–309. [79](#)
- [Chen et al., 2021b] Chen, Y., Wu, B., Jin, J., and Xu, K. (2021b). A variable curvature model for multi-backbone continuum robots to account for inter-segment coupling and external disturbance. *IEEE Robotics and Automation Letters*, 6(2) :1590–1597. [62](#), [63](#)
- [Chikhaoui and Rosa, 2022] Chikhaoui, M. T. and Rosa, B. (2022). Modeling and control strategies for flexible devices. pages 187–213. Elsevier. [8](#), [14](#), [78](#)
- [Childs and Rucker, 2023] Childs, J. A. and Rucker, C. (2023). A kinetostatic model for concentric push-pull robots. *IEEE Transactions on Robotics*. [14](#), [62](#), [63](#), [79](#)
- [Chun et al., 2010] Chun, K. J., Wissner, E., Koektuerk, B., Konstantinidou, M., Schmidt, B., Zerm, T., Metzner, A., Tilz, R., Boczor, S., Fuernkranz, A., et al. (2010). Remote-controlled magnetic pulmonary vein isolation using a new irrigated-tip catheter in patients with atrial fibrillation. *Circulation : Arrhythmia and Electrophysiology*, 3(5) :458–464. [IX](#), [7](#)
- [Cignolini et al., 2023] Cignolini, C., Rosa, B., and Courtecuisse, H. (2023). Fem-based model for cable-actuated continuum robots. In *12th Conference on New Technologies for Computer and Robot Assisted Surgery (CRAS 2023)*. [12](#)
- [Coad et al., 2019] Coad, M. M., Blumenschein, L. H., Cutler, S., Zepeda, J. A. R., Nacclerio, N. D., El-Hussieny, H., Mehmood, U., Ryu, J.-H., Hawkes, E. W., and Okamura, A. M. (2019). Vine robots. *IEEE Robotics & Automation Magazine*, 27(3) :120–132. [7](#)
- [Colgate et al., 1996] Colgate, J. E., Wannasuphoprasit, W., and Peshkin, M. A. (1996). Cobots : Robots for collaboration with human operators. In *ASME international mechanical engineering congress and exposition*, volume 15281, pages 433–439. American Society of Mechanical Engineers. [3](#)
- [Cosserat and Cosserat, 1909] Cosserat, E. and Cosserat, F. (1909). *Theory of deformable bodies*. Scientific Bookshop A. Hermann and Sons, Paris. [25](#), [45](#)
- [Cowin, 1962] Cowin, S. C. (1962). *Mechanics of Cosserat continua*. The Pennsylvania State University. [25](#)
- [Davies et al., 1991] Davies, B. L., Hibberd, R., Ng, W., Timoney, A., and Wickham, J. (1991). The development of a surgeon robot for prostatectomies. *Proceedings of the Institution of Mechanical Engineers, Part H : Journal of Engineering in Medicine*, 205(1) :35–38. [4](#)
- [Denavit and Hartenberg, 1955] Denavit, J. and Hartenberg, R. (1955). A kinematic notation for lower-pair mechanisms based on matrices. *Journal of Applied Mechanics*, 22(2) :215–221. [20](#)

- [Diezinger et al., 2022] Diezinger, M. A., Tamadazte, B., and Laurent, G. J. (2022). 3d curvature-based tip load estimation for continuum robots. *IEEE Robotics and Automation Letters*, 7(4) :10526–10533. [85](#)
- [Dong et al., 2022] Dong, X., Wang, M., Mohammad, A., Ba, W., Russo, M., Norton, A., Kell, J., and Axinte, D. (2022). Continuum robots collaborate for safe manipulation of high-temperature flame to enable repairs in challenging environments. *IEEE/ASME Transactions on Mechatronics*, 27(5) :4217–4220. [7](#)
- [Du et al., 2024] Du, F., Zhang, X., Wu, K., Zhang, G., Zheng, G., Li, Y., and Song, R. (2024). A sensor-free force estimation method for notched continuum surgical robot. *IEEE/ASME Transactions on Mechatronics*. [14](#), [84](#)
- [Dupont et al., 2009] Dupont, P. E., Lock, J., Itkowitz, B., and Butler, E. (2009). Design and control of concentric-tube robots. *IEEE Transactions on Robotics*, 26(2) :209–225. [8](#)
- [Dupont et al., 2022] Dupont, P. E., Simaan, N., Choset, H., and Rucker, C. (2022). Continuum robots for medical interventions. *Proceedings of the IEEE*, 110(7) :847–870. [1](#), [7](#)
- [Eto and Naito, 2005] Eto, M. and Naito, S. (2005). Robotic surgery assisted by the zeus system. *Endouroonology : New Horizons in Endourology*, pages 39–48. [IX](#), [4](#)
- [Fagogenis et al., 2019] Fagogenis, G., Mencattelli, M., Machaidze, Z., Rosa, B., Price, K., Wu, F., Weixler, V., Saeed, M., Mayer, J. E., and Dupont, P. E. (2019). Autonomous robotic intracardiac catheter navigation using haptic vision. *Science robotics*, 4(29) :eaaw1977. [14](#), [83](#)
- [Ferguson et al., 2024] Ferguson, J. M., Rucker, D. C., and Webster, R. J. (2024). Unified shape and external load state estimation for continuum robots. *IEEE Transactions on Robotics*. [14](#), [85](#), [90](#), [110](#)
- [Franco and Boyer, 2005] Franco, J.-S. and Boyer, E. (2005). Fusion of multiview silhouette cues using a space occupancy grid. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 1747–1753. IEEE. [XII](#), [102](#), [104](#)
- [Frazelle et al., 2020] Frazelle, C. G., Kapadia, A. D., and Walker, I. D. (2020). A haptic continuum interface for the teleoperation of extensible continuum manipulators. *IEEE Robotics and Automation Letters*, 5(2) :1875–1882. [83](#), [119](#)
- [Freundlich et al., 2015] Freundlich, C., Zavlanos, M., and Mordohai, P. (2015). Exact bias correction and covariance estimation for stereo vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3296–3304. [113](#)
- [Gao et al., 2019] Gao, Y., Takagi, K., Kato, T., Shono, N., and Hata, N. (2019). Continuum robot with follow-the-leader motion for endoscopic third ventriculostomy and tumor biopsy. *IEEE Transactions on Biomedical Engineering*, 67(2) :379–390. [14](#)
- [Girerd et al., 2020] Girerd, C., Kudryavtsev, A. V., Rougeot, P., Renaud, P., Rabenoro-soa, K., and Tamadazte, B. (2020). Slam-based follow-the-leader deployment of concentric tube robots. *IEEE Robotics and Automation Letters*, 5(2) :548–555. [14](#)

- [Girerd and Morimoto, 2020] Girerd, C. and Morimoto, T. K. (2020). Design and control of a hand-held concentric tube robot for minimally invasive surgery. *IEEE Transactions on Robotics*, 37(4) :1022–1038. [14](#)
- [Goldman et al., 2014] Goldman, R. E., Bajo, A., and Simaan, N. (2014). Compliant motion control for multisegment continuum robots with actuation force sensing. *IEEE Transactions on Robotics*, 30(4) :890–902. [83](#)
- [Ha et al., 2018] Ha, J., Fagogenis, G., and Dupont, P. E. (2018). Modeling tube clearance and bounding the effect of friction in concentric tube robot kinematics. *IEEE Transactions on Robotics*, 35(2) :353–370. [45](#), [118](#)
- [Ha et al., 2015] Ha, J., Park, F. C., and Dupont, P. E. (2015). Elastic stability of concentric tube robots subject to external loads. *IEEE Transactions on Biomedical Engineering*, 63(6) :1116–1128. [75](#)
- [Ha et al., 2016] Ha, J., Park, F. C., and Dupont, P. E. (2016). Optimizing tube precurvature to enhance the elastic stability of concentric tube robots. *IEEE Transactions on Robotics*, 33(1) :22–37. [78](#)
- [Hannan and Walker, 2000] Hannan, M. and Walker, I. (2000). Novel kinematics for continuum robots. In *Advances in Robot Kinematics*, pages 227–238. Springer. [8](#)
- [Hirose, 1983] Hirose, S. (1983). Tenser actuated elastic manipulator. In *in Proc. 6th IFToMM World Congress*, volume 2, pages 978–981. [6](#)
- [Ikuta et al., 1988] Ikuta, K., Tsukamoto, M., and Hirose, S. (1988). Shape memory alloy servo actuator system with electric resistance feedback and application for active endoscope. In *Proceedings. 1988 IEEE International Conference on Robotics and Automation*, pages 427–430. Ieee. [6](#)
- [Intuitive Surgical, 2024] Intuitive Surgical (2024). Systèmes chirurgicaux da vinci. Consulté le 4 octobre 2024. [IX](#), [5](#)
- [Itskov, 2000] Itskov, M. (2000). On the theory of fourth-order tensors and their applications in computational mechanics. *Computer Methods in Applied Mechanics and Engineering*, 189(2) :419–438. [119](#)
- [Jakes et al., 2019] Jakes, D., Ge, Z., and Wu, L. (2019). Model-less active compliance for continuum robots using recurrent neural networks. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2167–2173. IEEE. [118](#)
- [Janabi-Sharifi et al., 2021] Janabi-Sharifi, F., Jalali, A., and Walker, I. D. (2021). Cosserat rod-based dynamic modeling of tendon-driven continuum robots : A tutorial. *IEEE Access*, 9 :68703–68719. [62](#), [63](#)
- [Jang et al., 2016] Jang, C., Ha, J., Dupont, P. E., and Park, F. C. (2016). Toward on-line parameter estimation of concentric tube robots using a mechanics-based kinematic model. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2400–2405. IEEE. [109](#)
- [Jones and Walker, 2005] Jones, B. A. and Walker, I. D. (2005). A new approach to jacobian formulation for a class of multi-section continuum robots. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 3268–3273. IEEE. [9](#)

- [Jones and Walker, 2006] Jones, B. A. and Walker, I. D. (2006). Kinematics for multi-section continuum robots. *IEEE Transactions on Robotics*, 22(1) :43–55. [8](#)
- [Khadem et al., 2018] Khadem, M., Da Cruz, L., and Bergeles, C. (2018). Force/velocity manipulability analysis for 3d continuum robots. *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4920–4926. [52](#)
- [Khadem et al., 2019] Khadem, M., O’Neill, J., Mitros, Z., Da Cruz, L., and Bergeles, C. (2019). Autonomous steering of concentric tube robots for enhanced force/velocity manipulability. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2197–2204. IEEE. [14](#)
- [Korn et al., 2024] Korn, B., Jaded F, A., Enrico, F., and Ferdinando Rodriguez y, B. (2024). A pneumatic sensor for force measurement in colonoscopy. In *Hamlyn Symposium on Medical Robotics 2024*, pages 141–142. IEEE. [84](#)
- [Kwoh et al., 1988] Kwoh, Y. S., Hou, J., Jonckheere, E. A., and Hayati, S. (1988). A robot with improved absolute positioning accuracy for ct guided stereotactic brain surgery. *IEEE transactions on biomedical engineering*, 35(2) :153–160. [4](#)
- [Lee et al., 2017] Lee, K.-H., Leong, M. C., Chow, M. C., Fu, H.-C., Luk, W., Sze, K.-Y., Yeung, C.-K., and Kwok, K.-W. (2017). Fem-based soft robotic control framework for intracavitary navigation. In *2017 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, pages 11–16. IEEE. [12](#)
- [Levenberg, 1944] Levenberg, K. (1944). A method for the solution of certain non-linear problems in least squares. *Quarterly of applied mathematics*, 2(2) :164–168. [29](#)
- [Li et al., 2023] Li, H., Xun, L., and Zheng, G. (2023). Piecewise linear strain cosserat model for soft slender manipulator. *IEEE Transactions on Robotics*, 39(3) :2342–2359. [79](#)
- [Liang et al., 2021] Liang, N., Grassmann, R. M., Lilge, S., and Burgner-Kahrs, J. (2021). Learning-based inverse kinematics from shape as input for concentric tube continuum robots. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1387–1393. IEEE. [118](#)
- [Lin et al., 2019] Lin, N., Wu, P., Wang, M., Wei, J., Yang, F., Xu, S., Ye, Z., and Chen, X. (2019). Imu-based active safe control of a variable stiffness soft actuator. *IEEE Robotics and Automation Letters*, 4(2) :1247–1254. [83](#)
- [Lock and Dupont, 2011] Lock, J. and Dupont, P. E. (2011). Friction modeling in concentric tube robots. In *2011 IEEE International Conference on Robotics and Automation*, pages 1139–1146. IEEE. [45](#)
- [Lock et al., 2010] Lock, J., Laing, G., Mahvash, M., and Dupont, P. E. (2010). Quasi-static modeling of concentric tube robots with external loads. *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2325–2332. [11](#), [13](#), [19](#), [28](#), [45](#), [79](#)
- [Lou et al., 2010] Lou, L., Ramakrishna, K., Shao, L., Park, W.-T., Yu, D., Lim, L., Wee, Y., Kripesh, V., Feng, H., Chua, B. S., et al. (2010). Sensorized guidewires with mems tri-axial force sensor for minimally invasive surgical applications. In *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*, pages 6461–6464. IEEE. [84](#)

- [Mahoney et al., 2019] Mahoney, A. W., Gilbert, H. B., and Webster III, R. J. (2019). A review of concentric tube robots : modeling, control, design, planning, and sensing. *The Encyclopedia of Medical Robotics : Volume 1 Minimally Invasive Surgical Robotics*, pages 181–202. [IX, 6, 7](#)
- [Marescaux et al., 2001] Marescaux, J., Leroy, J., Gagner, M., Rubino, F., Mutter, D., Vix, M., Butner, S. E., and Smith, M. K. (2001). Transatlantic robot-assisted telesurgery. *Nature*, 413(6854) :379–380. [4](#)
- [MathWorks, 2023] MathWorks (2023). Matlab version : 9.14.0.2337262 (r2023a) update 5. [43](#)
- [Morecki et al., 1988] Morecki, A., Jaworek, K., Pogorzelski, W., Zielinska, T., Fraczek, J., and Malczyk, G. (1988). Robotics system—elephant trunk type elastic manipulator combined with a quadruped walking machine. In *Robotics and Factories of the Future'87 : Proceedings of the Second International Conference San Diego, California, USA July 28–31, 1987*, pages 649–656. Springer. [6](#)
- [Murray et al., 1994] Murray, R. M., Li, Z., and Sastry, S. S. (1994). *A mathematical introduction to robotic manipulation*. CRC press. [20, 47](#)
- [Neumann and Burgner-Kahrs, 2016] Neumann, M. and Burgner-Kahrs, J. (2016). Considerations for follow-the-leader motion of extensible tendon-driven continuum robots. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 917–923. IEEE. [8](#)
- [Newton, 1756] Newton, I. (1756). *Principes mathématiques de la philosophie naturelle*, volume 1. Desaint & Saillant. [13](#)
- [Noh et al., 2014] Noh, Y., Secco, E. L., Sareh, S., Würdemann, H., Faragasso, A., Back, J., Liu, H., Sklar, E., and Althoefer, K. (2014). A continuum body force sensor designed for flexible surgical robotics devices. In *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 3711–3714. IEEE. [84](#)
- [Oliver-Butler et al., 2021] Oliver-Butler, K., Childs, J. A., Daniel, A., and Rucker, D. C. (2021). Concentric push–pull robots : Planar modeling and design. *IEEE Transactions on Robotics*, 38(2) :1186–1200. [IX, 7](#)
- [Oliver-Butler et al., 2019] Oliver-Butler, K., Till, J., and Rucker, C. (2019). Continuum robot stiffness under external loads and prescribed tendon displacements. *IEEE Transactions on Robotics*, 35(2) :403–419. [11](#)
- [Olson, 2011] Olson, E. (2011). Apriltag : A robust and flexible visual fiducial system. In *2011 IEEE international conference on robotics and automation*, pages 3400–3407. IEEE. [102](#)
- [Open Continuum Robotics, 2022a] Open Continuum Robotics (2022a). Early continuum robotics history. Consulté le 7 octobre 2024. [IX, 6](#)
- [Open Continuum Robotics, 2022b] Open Continuum Robotics (2022b). Tensorarm history. Consulté le 7 octobre 2024. [IX, 6](#)
- [Orekhov et al., 2017] Orekhov, A. L., Aloï, V. A., and Rucker, D. C. (2017). Modeling parallel continuum robots with general intermediate constraints. *2017 IEEE International Conference on Robotics and Automation*, pages 6142–6149. [79](#)

- [Patel et al., 2022] Patel, R. V., Atashzar, S. F., and Tavakoli, M. (2022). Haptic feedback and force-based teleoperation in surgical robotics. *Proceedings of the IEEE*, 110(7) :1012–1027. [118](#)
- [Paul et al., 1992] Paul, H., Bargar, W., Mittlestadt, B., Kazanzides, P., Musits, B., Zuhars, J., Cain, P., Williamson, B., and Smith, F. (1992). Robotic execution of a surgical plan. In *IEEE International Conference on Systems, Man, and Cybernetics*, pages 1621–1623. IEEE. [4](#)
- [Penrose, 2007] Penrose, R. (2007). *A la découverte des lois de l’univers : La prodigieuse histoire des mathématiques et de la physique*. Odile Jacob. [13](#)
- [Peyron et al., 2022] Peyron, Q., Boehler, Q., Rougeot, P., Roux, P., Nelson, B. J., Andreff, N., Rabenorosa, K., and Renaud, P. (2022). Magnetic concentric tube robots : Introduction and analysis. *The International Journal of Robotics Research*, 41(4) :418–440. [75](#)
- [Peyron and Burgner-Kahrs, 2023] Peyron, Q. and Burgner-Kahrs, J. (2023). Stability analysis of tendon driven continuum robots and application to active softening. *IEEE Transactions on Robotics*. [118](#)
- [Pourafzal et al., 2021] Pourafzal, M., Talebi, H. A., and Rabenorosa, K. (2021). Piecewise constant strain kinematic model of externally loaded concentric tube robots. *Mechatronics*, 74 :102502. [14](#), [79](#)
- [Puangmali et al., 2011] Puangmali, P., Liu, H., Seneviratne, L. D., Dasgupta, P., and Althoefer, K. (2011). Miniature 3-axis distal force sensor for minimally invasive surgical palpation. *Ieee/Asme Transactions On Mechatronics*, 17(4) :646–656. [84](#)
- [Qian et al., 2019] Qian, L., Wu, J. Y., DiMaio, S. P., Navab, N., and Kazanzides, P. (2019). A review of augmented reality in robotic-assisted surgery. *IEEE Transactions on Medical Robotics and Bionics*, 2(1) :1–16. [118](#)
- [Qiao et al., 2021] Qiao, Q., Borghesan, G., De Schutter, J., and Vander Poorten, E. (2021). Force from shape—estimating the location and magnitude of the external force on flexible instruments. *IEEE Transactions on Robotics*, 37(5) :1826–1833. [14](#), [85](#), [109](#), [110](#)
- [Rao et al., 2021] Rao, P., Peyron, Q., Lilge, S., and Burgner-Kahrs, J. (2021). How to model tendon-driven continuum robots and benchmark modelling performance. *Frontiers in Robotics and AI*, 7 :630245. [8](#)
- [Rao et al., 2024] Rao, P., Salzman, O., and Burgner-Kahrs, J. (2024). Towards contact-aided motion planning for tendon-driven continuum robots. *IEEE Robotics and Automation Letters*. [83](#)
- [Renda et al., 2021] Renda, F., Messer, C., Rucker, C., and Boyer, F. (2021). A sliding-rod variable-strain model for concentric tube robots. *IEEE Robotics and Automation Letters*, 6(2) :3451–3458. [10](#), [14](#)
- [Riviere et al., 1997] Riviere, C. N., Rader, R. S., and Khosla, P. K. (1997). Characteristics of hand motion of eye surgeons. In *Proceedings of the 19th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. 'Magnificent Milestones and Emerging Opportunities in Medical Engineering' (Cat. No. 97CH36136)*, volume 4, pages 1690–1693. IEEE. [45](#)

- [Robinson and Davies, 1999] Robinson, G. and Davies, J. B. C. (1999). Continuum robots-a state of the art. In *Proceedings 1999 IEEE international conference on robotics and automation (Cat. No. 99CH36288C)*, volume 4, pages 2849–2854. IEEE. [6](#)
- [Robot Magazine, 2023] Robot Magazine (2023). Robot unimate et la révolution industrielle. Consulté le 7 octobre 2024. [IX](#), [3](#)
- [Rucker, 2011] Rucker, D. C. (2011). *The mechanics of continuum robots : model-based sensing and control*. PhD thesis, Vanderbilt University. [57](#)
- [Rucker et al., 2013] Rucker, D. C., Das, J., Gilbert, H. B., Swaney, P. J., Miga, M. I., Sarkar, N., and Webster, R. J. (2013). Sliding mode control of steerable needles. *IEEE Transactions on Robotics*, 29(5) :1289–1299. [47](#)
- [Rucker et al., 2010] Rucker, D. C., Jones, B. A., and Webster III, R. J. (2010). A geometrically exact model for externally loaded concentric-tube continuum robots. *IEEE transactions on robotics*, 26(5) :769–780. [11](#), [13](#), [19](#), [28](#), [45](#), [79](#)
- [Rucker and Webster, 2011] Rucker, D. C. and Webster, R. J. (2011). Computing jacobians and compliance matrices for externally loaded continuum robots. *2011 IEEE International Conference on Robotics and Automation*, pages 945–950. [9](#), [48](#), [49](#), [52](#), [53](#), [56](#), [57](#), [58](#), [62](#), [63](#), [77](#)
- [Rucker and Webster III, 2011] Rucker, D. C. and Webster III, R. J. (2011). Statics and dynamics of continuum robots with general tendon routing and external loading. *IEEE Transactions on Robotics*, 27(6) :1033–1044. [79](#)
- [Russo et al., 2023] Russo, M., Sadati, S. M. H., Dong, X., Mohammad, A., Walker, I. D., Bergeles, C., Xu, K., and Axinte, D. A. (2023). Continuum robots : An overview. *Advanced Intelligent Systems*, 5(5) :2200367. [1](#)
- [Ryu and Dupont, 2014] Ryu, S. C. and Dupont, P. E. (2014). Fbg-based shape sensing tubes for continuum robots. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3531–3537. IEEE. [85](#)
- [Sabetian et al., 2019] Sabetian, S., Looi, T., Diller, E. D., and Drake, J. (2019). Self-collision detection and avoidance for dual-arm concentric tube robots. *IEEE Robotics and Automation Letters*. [14](#)
- [Santiago et al., 2016] Santiago, J. L. C., Godage, I. S., Gonthina, P., and Walker, I. D. (2016). Soft robots and kangaroo tails : modulating compliance in continuum structures through mechanical layer jamming. *Soft Robotics*, 3(2) :54–63. [118](#)
- [Saudrais et al., 2024] Saudrais, C., Bayle, B., Vitrani, M.-A., and Vérité, F. (2024). Skin-stretch haptic feedback augmentation improves performance in a simulated laparoscopic palpation task. *IEEE Transactions on Haptics*. [119](#)
- [Scientific American, 1899] Scientific American (1899). Scientific american - january 21, 1899. *Scientific American*, 80(03) :43–43. [IX](#), [2](#)
- [Sestini et al., 2021] Sestini, L., Rosa, B., De Momi, E., Ferrigno, G., and Padoy, N. (2021). A kinematic bottleneck approach for pose regression of flexible surgical instruments directly from images. *IEEE Robotics and Automation Letters*, 6(2) :2938–2945. [118](#)

- [Sheridan, 1989] Sheridan, T. B. (1989). Telerobotics. *Automatica*, 25(4) :487–507. IX, 2
- [Shi et al., 2024] Shi, J., Abad, S.-A., Dai, J. S., and Wurdemann, H. A. (2024). Position and orientation control for hyperelastic multisegment continuum robots. *IEEE/ASME Transactions on Mechatronics*. 118
- [Smoljkic et al., 2014] Smoljkic, G., Reynaerts, D., Vander Sloten, J., and Vander Poorten, E. (2014). Compliance computation for continuum types of robots. *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1066–1073. 49, 77
- [Stan Institute, 2018] Stan Institute (2018). Histoire de la chirurgie robotique. Consulté le 7 octobre 2024. 1
- [Stöckel, 1998] Stöckel, D. (1998). Nitinol-a material with unusual properties. *Endovascular Update*, 1(1) :1–8. 24
- [Suzumori et al., 2023] Suzumori, K., Fukuda, K., Niiyama, R., and Nakajima, K. (2023). *The Science of Soft Robots : Design, Materials and Information Processing*. Springer Nature. IX, 7
- [Thompson, 2015] Thompson, K. M. (2015). *I, Robot : Nikola Tesla’s Telautomaton*. PhD thesis, UCLA. 2
- [Thuruthel et al., 2016] Thuruthel, T. G., Falotico, E., Cianchetti, M., and Laschi, C. (2016). Learning global inverse kinematics solutions for a continuum robot. In *ROMANSY 21-Robot Design, Dynamics and Control : Proceedings of the 21st CISM-IFTToMM Symposium, June 20-23, Udine, Italy*, pages 47–54. Springer. 12
- [Till et al., 2018] Till, J., Aloï, V., and Rucker, C. (2018). Real-time dynamics of soft and continuum robots based on cosserat rod models. *The International Journal of Robotics Research*, 38(6) :723–746. 79
- [Till et al., 2019] Till, J., Aloï, V., and Rucker, C. (2019). Real-time dynamics of soft and continuum robots based on cosserat rod models. *The International Journal of Robotics Research*, 38(6) :723–746. 10
- [Torres et al., 2012] Torres, L. G., Webster, R. J., and Alterovitz, R. (2012). Task-oriented design of concentric tube robots using mechanics-based models. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4449–4455. IEEE. 118
- [Troccaz, 2012] Troccaz, J. (2012). *Robotique médicale*. Lavoisier. 1
- [Tummers et al., 2024] Tummers, M., Boyer, F., Lebastard, V., Offermann, A., Troccaz, J., Rosa, B., and Chikhaoui, M. T. (2024). Continuum concentric push–pull robots : A cosserat rod model. *The International Journal of Robotics Research*. 6, 11, 14, 79
- [Tummers et al., 2023] Tummers, M., Lebastard, V., Boyer, F., Troccaz, J., Rosa, B., and Chikhaoui, M. T. (2023). Cosserat rod modeling of continuum robots from newtonian and lagrangian perspectives. *IEEE Transactions on Robotics*. 14, 62, 63, 79
- [Universal Robots, 2024] Universal Robots (2024). Ur30 robot. Consulté le 7 octobre 2024. IX, 3
- [University of Toronto - CRL, 2021] University of Toronto - CRL (2021). Tactile data collection and representation. Consulté le 7 octobre 2024. IX, 7

- [Veneziano et al., 2020] Veneziano, D., Tafuri, A., Rivas, J. G., Dourado, A., Okhunov, Z., Somani, B., Marino, N., Fuchs, G., Cacciamani, G., and Group, E.-Y. (2020). Is remote live urologic surgery a reality? evidences from a systematic review of the literature. *World Journal of Urology*, 38 :2367–2376. [IX, 4](#)
- [Wang et al., 2023a] Wang, F., Jiang, Q., and Li, J. (2023a). Shape sensing for continuum robots using fbg sensors array considering bending and twisting. *IEEE Sensors Journal*. [85](#)
- [Wang et al., 2023b] Wang, N., Zhang, X., Stoyanov, D., Zhang, H., and Stilli, A. (2023b). Vision-and-force-based compliance control for a posterior segment ophthalmic surgical robot. *IEEE Robotics and Automation Letters*. [48](#)
- [Wang et al., 2024] Wang, Y., McCandless, M., Donder, A., Pittiglio, G., Moradkhani, B., Chitalia, Y., and Dupont, P. E. (2024). Using neural networks to model hysteretic kinematics in tendon-actuated continuum robots. In *2024 International Symposium on Medical Robotics (ISMR)*, pages 1–7. [118](#)
- [Webster III and Jones, 2010] Webster III, R. J. and Jones, B. A. (2010). Design and kinematic modeling of constant curvature continuum robots : A review. *The International Journal of Robotics Research*, 29(13) :1661–1683. [8](#)
- [Wilson, 1985] Wilson, J. (1985). Compliant robotic structures, part i. *School of Engineering, Duke University, Tech. Rep.* [6](#)
- [Wooten et al., 2018] Wooten, M., Frazelle, C., Walker, I. D., Kapadia, A., and Lee, J. H. (2018). Exploration and inspection with vine-inspired continuum robots. In *2018 IEEE International conference on robotics and automation (ICRA)*, pages 5526–5533. IEEE. [7](#)
- [Wu and Zheng, 2022] Wu, K. and Zheng, G. (2022). Fem-based nonlinear controller for a soft trunk robot. *IEEE Robotics and Automation Letters*, 7(2) :5735–5740. [12](#)
- [Xiang et al., 2023] Xiang, P., Zhang, J., Sun, D., Qiu, K., Fang, Q., Mi, X., Wang, Y., Xiong, R., and Lu, H. (2023). Learning-based high-precision force estimation and compliant control for small-scale continuum robot. *IEEE Transactions on Automation Science and Engineering*. [118](#)
- [Xie et al., 2022] Xie, M., Girerd, C., and Morimoto, T. K. (2022). A 3-d haptic trackball interface for teleoperating continuum robots. In *2022 9th IEEE RAS/EMBS International Conference for Biomedical Robotics and Biomechatronics (BioRob)*, pages 1–8. IEEE. [83](#)
- [Xu and Simaan, 2008] Xu, K. and Simaan, N. (2008). An investigation of the intrinsic force sensing capabilities of continuum robots. *IEEE Transactions on Robotics*, 24(3) :576–587. [85](#)
- [Xu et al., 2013] Xu, R., Asadian, A., Naidu, A. S., and Patel, R. V. (2013). Position control of concentric-tube continuum robots using a modified jacobian-based approach. *2013 IEEE International Conference on Robotics and Automation*, pages 5813–5818. [9, 52](#)
- [Xu and Patel, 2012] Xu, R. and Patel, R. V. (2012). A fast torsionally compliant kinematic model of concentric-tube robots. In *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 904–907. IEEE. [28](#)

- [Xu et al., 2017] Xu, W., Chen, J., Lau, H. Y., and Ren, H. (2017). Data-driven methods towards learning the highly nonlinear inverse kinematics of tendon-driven surgical manipulators. *The International Journal of Medical Robotics and Computer Assisted Surgery*, 13(3) :e1774. [12](#)
- [Yang et al., 2023] Yang, Z., Yang, H., Cao, Y., Cui, Y., and Zhang, L. (2023). Magnetically actuated continuum medical robots : A review. *Advanced intelligent systems*, 5(6). [6](#)
- [Yasin and Simaan, 2021] Yasin, R. and Simaan, N. (2021). Joint-level force sensing for indirect hybrid force/position control of continuum robots with friction. *The International Journal of Robotics Research*, 40(4-5) :764–781. [14](#)
- [Yip et al., 2010] Yip, M. C., Yuen, S. G., and Howe, R. D. (2010). A robust uniaxial force sensor for minimally invasive surgery. *IEEE transactions on biomedical engineering*, 57(5) :1008–1011. [14](#)
- [Yoshikawa, 1985] Yoshikawa, T. (1985). Manipulability of robotic mechanisms. *The international journal of Robotics Research*, 4(2) :3–9. [119](#)
- [Zeng et al., 2021] Zeng, W., Yan, J., Yan, K., Huang, X., Wang, X., and Cheng, S. S. (2021). Modeling a symmetrically-notched continuum neurosurgical robot with non-constant curvature and superelastic property. *IEEE Robotics and Automation Letters*, 6(4) :6489–6496. [118](#)
- [Zhao et al., 2022] Zhao, X., Li, J., Zhou, Y., and Chen, J. (2022). A soft haptic interface for programmable patterns of touch. *Matter*, 5(9) :2590–2593. [119](#)
- [Čapek, 1920] Čapek, K. (1920). *R.U.R. (Rossum’s Universal Robots)*. [2](#)

Modélisation et utilisation de la compliance des robots continus

Application aux robots à tubes concentriques

Résumé :

Initialement développés pour automatiser les industries manufacturières, les robots sont désormais utilisés pour réaliser des interventions chirurgicales. Avec la volonté de minimiser le traumatisme des patients, les robots continus, flexibles et hautement miniaturisables, sont devenus un sujet de recherche très actif. La plupart des algorithmes de commande de ces robots utilisent des modèles analytiques construits à partir de la théorie de Cosserat. Cette thèse reconsidère le concept de compliance des robots continus en proposant une nouvelle définition : la matrice de compliance généralisée. Cette dernière est calculée grâce à une méthode littérale dite de "propagation de dérivées bas-niveau", spécialement conçue pour permettre un calcul rapide. Ce concept de compliance généralisée est ensuite utilisé pour estimer les efforts d'interaction entre le robot et son environnement, élément indispensable pour garantir la sûreté de l'intervention. Tout au long de cette thèse, les contributions développées pour les robots continus sont systématiquement appliquées au cas particulier des robots à tubes concentriques, à travers des simulations et des essais expérimentaux.

Mots clés— Robots continus, Robots à tubes concentriques, Théorie de Cosserat, Compliance généralisée, Méthode par propagation de dérivées, Estimation d'efforts

Abstract :

First developed to automate production in manufacturing industries, robots are now used to assist surgical procedures. Flexible and highly miniaturizable continuum robots, allowing for the reduction of patient trauma, have raised significant interest in this field. Their study has then become an active research field. Most control algorithms developed for these robots are based on literal models derived from the Cosserat theory. This thesis reconsiders the concept of compliance for continuum robots by introducing the concept of generalized compliance matrix. This matrix is computed using a "low-level derivative propagation method" specifically designed for efficient computation. This concept of generalized compliance is then applied to estimate the interaction forces between the robot and its environment, a crucial issue for guaranteeing safe interventions. In this thesis, the contributions described for continuum robots are systematically transferred to the concentric tubes robots through simulations and experimental studies.

Keywords— Continuum robots, Concentric tubes robots, Cosserat theory, Generalized Compliance, Derivative propagation method, Force sensing