



HAL
open science

Manipulation robotique d'objets déformables basée sur le modèle Cosserat

Azad Artinian

► **To cite this version:**

Azad Artinian. Manipulation robotique d'objets déformables basée sur le modèle Cosserat. Robotique [cs.RO]. Sorbonne Université, 2024. Français. NNT : 2024SORUS080 . tel-04633245

HAL Id: tel-04633245

<https://theses.hal.science/tel-04633245>

Submitted on 3 Jul 2024

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

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

THÈSE de DOCTORAT

pour obtenir le titre de

Docteur de Sorbonne Université
École doctorale Sciences Mécaniques, Acoustique,
Électronique et Robotique de Paris
Mention : ROBOTIQUE

Présentée et soutenue par AZAD ARTINIAN

Manipulation robotique d'objets déformables basée sur le modèle Cosserat

soutenance prévue le 01/02/2024

Jury :

<i>Rapporteurs :</i>	Christian DURIEZ	- Directeur de recherche, INRIA
	Mathieu GROSSARD	- Directeur de recherche, CEA
<i>Examineurs :</i>	Youcef MEZOUAR	- Professeur des universités, SIGMA
	Brahim TAMADAZTE	- Directeur de Recherche, ISIR
<i>Directeurs :</i>	Véronique PERDEREAU	- Professeure des universités, ISIR
	Faïz BEN AMAR	- Professeur des universités, ISIR

Abstract :

La manipulation d'objets déformables fait partie des tâches évidentes pour un être humain. Nous en manipulons tous les jours, que ce soient des habits, de la nourriture ou des accessoires.

Même pour des objets que nous n'avons jamais vu auparavant, il est intuitif de savoir à l'avance comment l'objet va réagir et se déformer lorsqu'on le manipule. En voyant uniquement la géométrie, l'aspect et la texture d'un objet, et en se référant à nos expériences passées avec d'autres objets similaires, on peut très souvent prédire à l'avance la manière dont l'objet va se déformer sans même le toucher. De plus, il nous est très facile, avec un nombre réduit d'interactions avec un nouveau matériau, d'extrapoler et de déduire rapidement l'ensemble des actions à effectuer pour déformer l'objet spécifiquement.

Les robots ne possèdent toutefois pas cette mémoire pour les aider à appréhender de nouvelles situations. Pour eux, il n'existe aucun lien évident entre les actions qu'ils effectuent sur un objet et la forme que prend celui-ci. Permettre aux robots de manipuler des objets déformables est donc un défi en robotique. Derrière l'apparente simplicité de la tâche, retranscrire ce lien que nous possédons intuitivement entre les actions effectuées et la forme d'un objet dans un langage compréhensible pour un robot est un problème complexe. Cependant, de nombreuses tâches impliquant de manipuler et de traiter des objets déformables bénéficieraient grandement de l'aide de robots, que ce soit dans les secteurs industriels, publics ou même dans la vie de tous les jours.

Dans cette thèse, nous cherchons à faire avancer la recherche dans ce secteur en proposant une approche permettant à un robot de manipuler des objets déformables. En se basant sur un modèle mécanique de l'objet, nous proposons dans un premier temps une méthode hors-ligne pour planifier les actions du robot de sorte à réaliser une tâche de manipulation d'objet déformable en prenant en compte les contraintes appliquées sur l'objet. Dans un second temps, nous étendons l'approche précédente à une tâche de détachement d'un objet déformable fixé sur une surface plane. Enfin, nous proposons une méthode de contrôle de forme en ligne, qui peut être utilisée conjointement avec la méthode de planification ou indépendamment.

Remerciements

Je tiens à remercier profondément l'ensemble des personnes qui m'ont aidé durant cette thèse.

Merci à mes amis Dounia, Ali, Loïc, Ghani et Elias, votre compagnie a souvent été ma principale motivation pour venir au laboratoire, je ne pense pas que j'y serai arrivé sans vous.

Merci à Basma de m'avoir épaulé et accompagné ces deux dernières années.

Merci à ma famille de m'avoir soutenu et encouragé, notamment pendant le covid.

Enfin, merci à mes directeurs de thèse Faïz et Véronique qui m'ont accordé leur confiance et permis de réaliser cette thèse.

Table des matières

Liste d'abréviations	ix
Support Video	xi
1 Introduction	1
1.1 Contexte	1
1.1.1 Applications	1
1.1.2 Difficultés à manipuler des objets déformables	2
1.2 Projet Softmanbot	3
1.3 Cadre	6
1.4 Contributions	8
1.5 Publications	9
2 Etat de l'art général sur la modélisation des objets déformables pour la manipulation en robotique	11
2.1 Caractérisation et classification des méthodes de modélisation	11
2.1.1 Modèles discrets	14
2.1.2 Modèles physiquement réalistes	17
2.1.3 Approches basées données	21
2.2 Modèle Cosserat	23
2.2.1 Applications robotiques - cinématique et dynamique . .	23
2.2.2 Solution optimale	27
2.2.3 Formulation Lagrangienne vs Newtonienne	28
2.3 Cadre mathématique du modèle Cosserat	30
2.3.1 Cinématique	31
2.3.2 Equilibre statique	33
2.3.3 Sens physique des taux de variation	34
2.3.4 Loi de comportement	34
2.3.5 Equations de comportement	36
3 Génération de trajectoire pour la manipulation d'un objet déformable linéaire	37
3.1 Introduction et objectifs	37
3.1.1 Cadre de contrôle de robots déformables	37
3.1.2 Objectif	38
3.2 Formulation du problème à objectifs multiples	40
3.2.1 Forme désirée	40

3.2.2	Énergie de déformation	42
3.2.3	Fonction de coût	44
3.3	Résolution numérique et génération de trajectoire	45
3.3.1	Méthode de résolution	45
3.3.2	Minima locaux	46
3.3.3	Génération de trajectoires	48
3.4	Validation expérimentale	51
3.4.1	Dispositif Expérimental	51
3.4.2	Estimation des constantes élastiques	53
3.4.3	Répétabilité	55
3.4.4	Modélisation d'objets planaires avec le modèle Cosserat	57
3.4.5	Discussion et conclusions	58
4	Approche globale et robustesse de la méthode	63
4.1	Problème multiobjectif et recherche globale	64
4.2	Application à la tâche de détachement d'un objet planaire	66
4.2.1	Introduction et objectifs de la partie	66
4.2.2	Présentation de la maquette	68
4.2.3	Modélisation	69
4.2.4	Contrôle en impédance	71
4.2.5	Détachement de la bande élastique	74
4.3	Conclusion	76
5	Contrôle de la déformation en boucle fermée	79
5.1	Modèle Cosserat	82
5.2	Contrôle de forme	82
5.2.1	Formulation du problème	82
5.2.2	Matrice Jacobienne des déformations basée sur le modèle Cosserat	83
5.2.3	Commande	84
5.2.4	Perception	85
5.2.5	Contrôle en boucle fermée	87
5.2.6	Problème bien posé et critère de stabilité	87
5.3	Validation expérimentale	89
5.3.1	Dispositif expérimental	89
5.3.2	Protocole et configuration initiale de l'objet	91
5.3.3	Manipulation à un bras	93
5.3.4	Manipulation à deux bras	95
5.3.5	Résultats pour différents objets complexes	96
5.3.6	Configurations instables et temps de convergence	99
5.3.7	Répétabilité	100

Table des matières	vii
5.4 Conclusion et perspectives	101
6 Conclusion	105
6.1 Récapitulatif	105
6.2 Limitations et perspectives	107
Bibliographie	111

Liste d'abréviations

MSD : Mass-Spring-Damper
PBD : Point Based Dynamics
ARAP : As-Rigid-As-Possible
FEM : Finite Element Method
BVP : Boundary Value Problem
IVP : Initial Value Problem

Support Video

Chapitre 3 :

Trajectoire robotique pour la manipulation d'objets linéaires :
<https://www.youtube.com/watch?v=EWHvZS0bhW8>

Chapitre 4 :

Expérience de détachement d'une bande élastique : <https://www.youtube.com/watch?v=Hq2mJ0W29LI>

Chapitre 5 :

Contrôle de forme à deux bras en boucle fermée - objet 1 :
<https://www.youtube.com/watch?v=8s7yjiQMLs0>

Contrôle de forme à deux bras en boucle fermée - objet 2 :
<https://www.youtube.com/watch?v=RKaIdVLLVPE>

Contrôle de forme à deux bras en boucle fermée - objet 3 :
<https://www.youtube.com/watch?v=X0m77ZQ6eFQ>

Contrôle de forme à une bras en boucle fermée : <https://www.youtube.com/watch?v=DR7LUhRHN-E>

CHAPITRE 1

Introduction

Dans ce chapitre, on introduira le contexte dans lequel est réalisé cette thèse, les applications liées à la manipulation robotique d'objets déformables ainsi que les difficultés liées à ce sujet. On présentera également le projet Softmanbot dans le cadre duquel cette thèse a été réalisée. Le projet européen Softmanbot a pour ambition de développer de nouvelles solutions robotiques pour la manipulation d'objets déformables dans un contexte industriel. On détaillera les cas d'études du projet qui couvrent un ensemble varié de tâches et objets différents. On posera ensuite les définitions et hypothèses faites au cours de la suite.

1.1 Contexte

On pose ici le contexte général pour la manipulation robotique d'objets déformables. Plus spécifiquement, les applications potentielles dans l'industrie et dans la vie de tous les jours ainsi que les difficultés spécifiques du sujet.

1.1.1 Applications

Les objets déformables sont omniprésents, que ce soit dans notre vie de tous les jours ou dans l'industrie. Arriver à automatiser leur manipulation est un enjeu crucial dans l'ensemble des principaux secteurs industriels. On en trouve sous forme de câbles dans l'industrie automobile, aéronautique et aérospatiale, mais également dans les secteurs médicaux et chirurgicaux où le caractère déformable de tubes et tiges permet d'atteindre et de traiter des parties du corps humain difficilement atteignables autrement. L'industrie plasturgique en est également remplie, ce secteur rassemble l'ensemble des techniques utilisées par l'industrie du plastique pour concevoir ou transformer les matières plastiques. Le constat est le même pour l'industrie du textile et l'industrie agro-alimentaire qui traitent quasiment exclusivement des objets déformables. Automatiser des tâches qui pourraient être dangereuses à court et à long termes si réalisées par des opérateurs humains, assister l'humain pour accomplir des tâches complexes, par exemple pour une opération chirurgicale, ainsi qu'augmenter la production dans ces secteurs sont des perspectives primordiales pour de nombreuses entreprises et établissements publics.

La manipulation robotique d'objets déformables a donc suscité un intérêt important dans la communauté robotique ces dernières années.

1.1.2 Difficultés à manipuler des objets déformables

La manipulation robotique d'objets déformables reste aujourd'hui un problème ouvert et les méthodes applicables hors d'un laboratoire sont encore rares. Bien que de nombreuses méthodes aient été développées pour manipuler des d'objets rigides, celles-ci sont très rarement applicables aux objets déformables. Ces derniers ont un nombre infini de degrés de liberté et bien qu'il soit intuitif pour un humain de les manipuler, cela reste un problème complexe pour un robot. En effet, dans le cas des objets rigides, les points de l'objet manipulé sont reliés entre eux par une transformation rigide. En appliquant les conditions de fermeture en force et/ou en forme, deux conditions généralement utilisées dans le domaine de la saisie robotique, on s'assure d'avoir une prise stable. Pour pouvoir contrôler chaque point d'un objet rigide, il suffit ainsi de calculer la transformation rigide entre ce point et le préhenseur.

Cette approche n'est typiquement pas applicable aux objets déformables. La condition de fermeture en forme [Bicchi 1995], permettant de fixer l'objet par rapport au robot, n'est pas applicable à des objets susceptibles de se déformer sous l'action du préhenseur [Guo 2013]. La condition de fermeture en force peut être appliquée, mais doit être recalculée en permanence pour prendre en compte le changement d'inertie lorsque l'objet se déforme [Jia 2011, Jia 2013]. Même en considérant un contact parfait entre le préhenseur et l'objet, le lien entre les actions du robot et la forme résultante de l'objet n'est pas trivial.

De nombreuses approches dans la littérature ont tenté de modéliser ce lien, que ce soit à l'aide de modèles mécaniques, modèles géométriques ou modèles appris. Pour contrôler précisément un objet déformable, il est toutefois nécessaire d'inverser ce lien, de sorte à obtenir le lien inverse donnant les actions du robot à effectuer pour déformer l'objet dans une forme désirée. Réaliser une tâche de manipulation implique ensuite de générer la trajectoire de l'objet, d'où on extrait celle du robot.

Dans cette thèse on se concentrera principalement sur les trois points suivants : modélisation, planification et contrôle de forme. Les autres aspects nécessaires à la manipulation d'objets déformables à savoir la perception, la saisie et la conception du robot et du préhenseur sont représentés sur la figure 1.1.

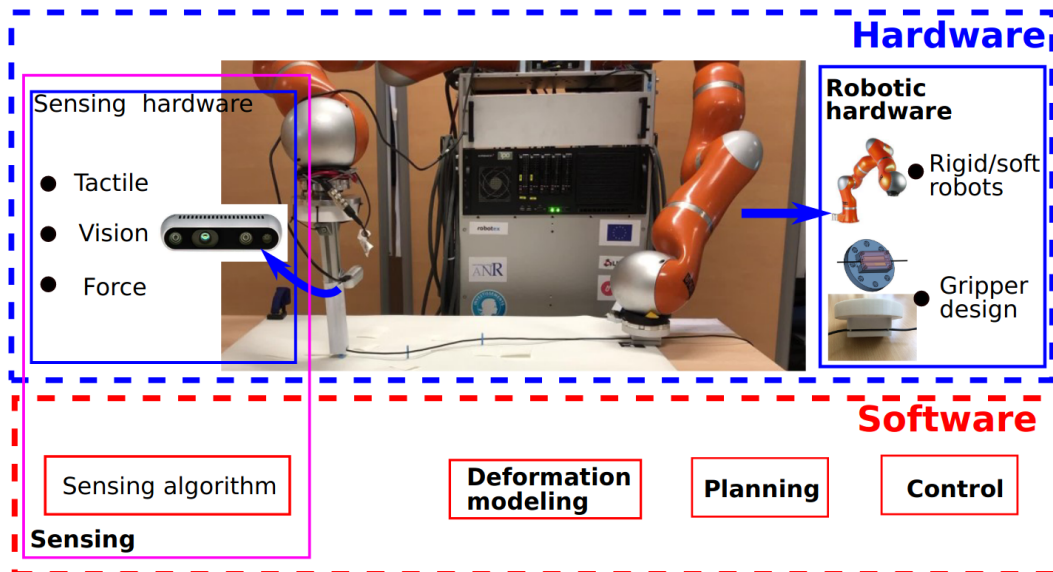


FIGURE 1.1 – Ensemble des aspects nécessaires à la manipulation robotique d’objets déformables [Zhu 2022]

1.2 Projet Softmanbot

Cette thèse a été réalisée dans le cadre du projet européen SOFTMANBOT. Le projet SOFTMANBOT est un projet européen à visée industrielle dont le but est de développer de nouvelles solutions pour la manipulation robotique d’objets déformables dans des contextes industriels. Le projet a pour ambition d’innover dans trois secteurs, à savoir la perception des déformations, la planification et le contrôle multi-sensoriel ainsi que la conception de préhenseurs. Les solutions proposées ont pour but de fonctionner en collaboration avec les opérateurs humains et de les assister pour accomplir des tâches de manipulation afin d’augmenter non seulement l’ergonomie des postes de travail, mais également la production. Le but final est de ramener, ou conserver la production industrielle en Europe.

Les solutions développées sont ensuite testées dans quatre cas d’études qui représentent des secteurs industriels clés (du jouet, du textile, de la chaussure, du pneu) :

- **Cas d’étude Juema 1** : Extraction de parties de poupées du moule (figure 1.2).
- **Cas d’étude Juema 2** : Assemblage de parties de poupées (figure 1.3).
- **Cas d’étude Plastinher** : Extraction de semelles du moule (figure 1.4).
- **Cas d’étude Michelin** : Assemblage de pneu (figure 1.5).

- **Cas d'étude Decathlon** : Assemblage et couture d'un pad en mousse et d'un textile (figure 1.6).

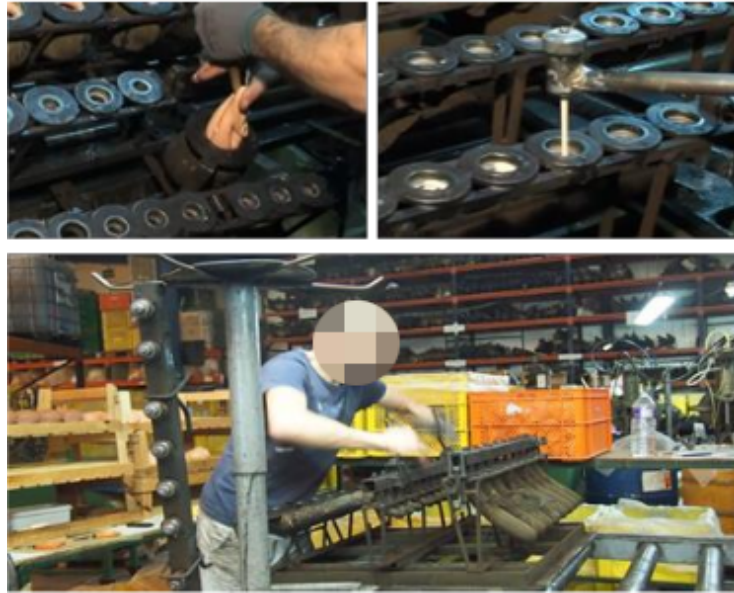


FIGURE 1.2 – Démoulage manuel de parties de poupées par un opérateur - cas d'étude Juema



FIGURE 1.3 – Assemblage manuel de parties de poupées par un opérateur - cas d'étude Juema 2

Une méthode permettant de contrôler en boucle fermée la forme d'un objet avec un taux de succès de 90 pour cent pour une tâche précise est très impressionnant en terme de recherche, mais d'un point de vue industriel, perdre 10 pour cent des matériaux est difficilement envisageable. Jusqu'ici, il n'existe pas dans la littérature d'approche permettant de réaliser une tâche de manipulation tout en prenant en compte les contraintes liées à une application et à un environnement industriel.

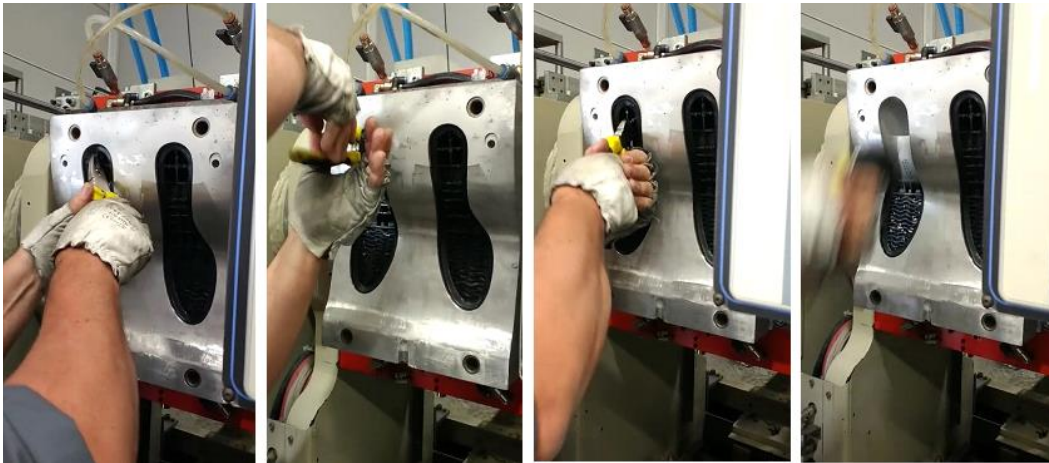


FIGURE 1.4 – Processus manuel de démoulage de semelles - cas d'étude Plastinher



FIGURE 1.5 – Processus d'assemblage de pneus - cas d'étude Michelin

Afin de faire face aux difficultés soulevées par les cas d'étude du projet Softmanbot, il a donc été nécessaire de concevoir une nouvelle méthode permettant de générer une trajectoire robotique qui permette de réaliser une tâche de manière précise, répétable, et sans endommager l'objet. Ce dernier point est particulièrement important car de nombreuses applications industrielles de manipulation d'objets déformables impliquent des matières fragiles comme le caoutchouc ou le plastique chaud.

Ces matières ont une plage élastique qui peut être très faible, surtout à haute température, et il suffit que le robot applique des efforts au dessus d'une certaine limite pour étirer et endommager irrémédiablement le matériau. Dans certains cas comme avec le plastique, le matériau endommagé peut être



FIGURE 1.6 – Assemblage manuel puis couture d’un pad en mousse sur un textile - cas d’étude Decathlon

réchauffé et ré-inclus dans le processus, mais dans d’autres cas, notamment pour des matériaux composites, ce n’est pas possible et les pièces abîmées finissent à la poubelle.

Outre la fragilité du matériau, les efforts nécessaires lors d’applications comme le démoulage sont particulièrement élevés, trouver une trajectoire qui permette à la fois de réaliser la tâche sans endommager l’objet est donc une réelle difficulté, même pour un opérateur humain. Par exemple, pour le cas du démoulage de parties de poupées (voir figure 1.2), l’intégralité des pièces démoulées manuellement subissent un étirement compris entre 10 et 50 pour cent de leur taille réelle et doivent donc être systématiquement réchauffées pour retrouver leur taille originale.

1.3 Cadre

Les objets déformables sont une famille d’objets extrêmement vaste. Une orange, un câble métallique tressé et une serviette sont trois exemples d’objets déformables. Ces exemples ont tous une topologie complètement différente, sont manipulés de manière différente et dans des contextes différents. On adoptera ainsi la classification proposée par [Sanchez 2018] qui différencie les objets en fonction de leurs propriétés physiques et de leur géométrie, cette classification est illustrée sur la figure 2.1. On utilisera également la définition de Sanchez d’un objet déformable : objet qui peut se déformer tout en maintenant sa topologie. Cette définition a l’avantage d’exclure l’ensemble des agrégats tels que le sable, ainsi que les fluides non newtoniens.

Dans cette thèse, on a choisi de restreindre notre étude aux objets linéaires, c’est-à-dire aux objets qui ont une dimension très grande devant les deux autres tels que des cordes, câbles et poutres. Ce choix est motivé par la forte

présence dans l'industrie de ce type d'objet, ainsi que par les méthodes utilisées pour les manipuler.

De plus, tout au long de cette thèse, on fera les hypothèses suivantes :

- Les objets manipulés ont un comportement élastique (les différents types de déformations sont illustrés sur la figure 1.7).
- Les effets liés à la viscosité de ces objets sont négligeables.
- Le contact entre le préhenseur du robot et l'objet est parfait, c'est-à-dire qu'il n'y a ni glissements ni rotations au point de contact au cours de la manipulation.
- Hypothèse quasi-statique : les phénomènes inertiels sont négligés pendant la manipulation.

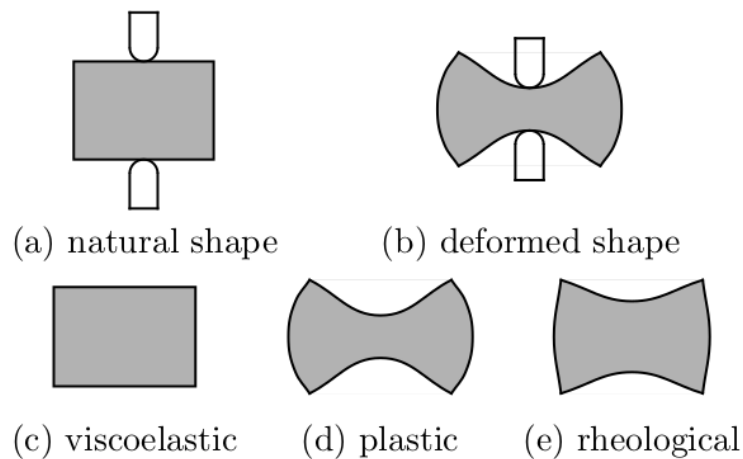


FIGURE 1.7 – Types de déformations, proposés par les auteurs de [Tokumoto 2001]

La plupart des approches développées pour la manipulation d'objets déformables dans la littérature se concentrent sur le contrôle de forme exclusivement. Il existe des approches basées données permettant de générer une trajectoire pour réaliser une tâche complète. Bien que celles-ci donnent des résultats satisfaisants, leur robustesse et leur taux de succès ne sont pas encore assez élevés pour être réellement intéressants pour l'industrie. A notre connaissance, la totalité des méthodes permettant de réaliser une tâche de manipulation d'objets déformables dans la littérature se basent sur des hypothèses difficiles à satisfaire sous des contraintes industrielles. Aucune n'a alors encore été appliquée dans un tel environnement.

Résoudre ces problèmes ne peut ainsi pas être ainsien intégrant des méthodes pré-existantes, il a donc été nécessaire d'en développer de nouvelles en prenant en compte le contexte d'application dès le début.

1.4 Contributions

Dans cette thèse, on va dans un premier temps proposer une méthode permettant de résoudre une tâche de manipulation d'objet déformable linéaire de manière robuste, répétable, précise et adaptée à un environnement industriel. On modélisera l'objet manipulé à l'aide de la théorie de Cosserat. On s'inspirera pour cela du cadre proposé dans la communauté du contrôle de robots déformables qu'on va ensuite reformuler pour l'adapter à la manipulation d'objets linéaires. Afin de s'assurer que la méthode permette de prendre en compte les contraintes appliquées à l'objet, on couplera le cadre reformulé précédemment à la théorie du contrôle optimal. On définira ainsi un double objectif : un objectif en position, représentant la forme que l'on cherche à atteindre, et un objectif de minimisation de l'énergie élastique dans l'objet. On résoudra le problème d'optimisation à l'aide de la méthode de tir, puis on définira un ensemble d'objectifs intermédiaires. On résoudra chaque problème intermédiaire de manière itérative afin d'obtenir une trajectoire optimale de l'objet. Cette trajectoire sera ensuite testée pour une application de contrôle de forme d'objet linéaire déformable.

Dans un second temps, on apportera des modifications à cette approche en remplaçant l'algorithme d'optimisation local utilisé précédemment par un algorithme de résolution global. On proposera ensuite de remplacer le contrôleur en position du robot par un contrôleur en impédance. Ce changement de contrôleur permettra de rendre l'approche plus robuste aux conditions initiales, ainsi qu'aux phénomènes non modélisés. Enfin, on adaptera notre approche pour résoudre un problème similaire au problème de démoulage de semelle, qui correspond au cas d'étude Plastinher (figure 1.4).

Enfin, on travaillera sur un contrôleur en boucle fermée, basé sur le modèle Cosserat, qui pourra être utilisé pour atteindre la configuration définie avec l'approche précédente, ou de manière indépendante. Ce contrôleur permettra de contrôler la forme d'un objet linéaire avec une précision inférieure à 1mm, sans nécessiter de connaissance précise des paramètres élastiques de l'objet. Pour cela, on calculera, à chaque itération, une Jacobienne locale des déformations. Celle-ci est calculée à partir du modèle Cosserat et des données visuelles estimant la configuration actuelle de l'objet. A partir de cette Jacobienne, on mettra à jour directement les valeurs initiales du problème Cosserat. En résolvant le problème aux valeurs limites, on obtient ainsi à chaque itération une configuration physiquement réaliste et mise à jour de l'objet, jusqu'à la convergence vers la forme souhaitée. L'algorithme sera ensuite testé pour une application de contrôle de forme d'objet linéaire, manipulé avec deux bras robotiques. Pour montrer la robustesse de la méthode à l'estimation des paramètres élastiques, on manipulera des objets complexes,

non homogènes, non isotropes tels que des câbles métalliques tressés et gainés.

1.5 Publications

Ce travail a donné lieu aux publications suivantes :

Azad, A., Quentin, H., Faiz, B. A., Véronique, P. (2023, June). Optimal Cosserat-based deformation control for robotic manipulation of linear objects. In 2023 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM) (pp. 381-388). IEEE.

(A venir) Artinian, A., Ben Amar, F., Perdereau, V. Closed loop controller based on Cosserat theory for bi-arm manipulation of complex linear objects

Etat de l'art général sur la modélisation des objets déformables pour la manipulation en robotique

Le spectre des approches développées pour la manipulation d'objets déformables est très étendu. Dans un premier temps, on tentera de classifier et de faire ressortir les avantages et inconvénients de chaque type de méthodes. A travers cet état de l'art, on justifiera ainsi le choix de l'utilisation du modèle de poutre de Cosserat, utilisé dans la suite de cette thèse. On détaillera ensuite plus spécifiquement ce modèle et on posera son cadre mathématique en détail.

2.1 Caractérisation et classification des méthodes de modélisation

Contrôler la déformation d'un objet déformable implique de connaître à l'avance les effets de la commande du robot sur la forme de l'objet. Capturer de manière précise cette connexion entre les actions du robot et la déformation de l'objet qui en résulte est crucial pour réaliser une tâche de manipulation. La quasi totalité des approches ont tenté de modéliser ce lien, que ce soit en l'apprenant, en l'estimant, ou en utilisant un modèle pré-établi. L'ensemble des méthodes utilisées est extrêmement vaste. Afin de définir l'approche la plus adaptée, il est crucial de répertorier et classifier cet ensemble afin d'identifier les avantages, inconvénients et domaine d'application de chaque méthode. De nombreux auteurs ont proposé leur classification. Dans [Jiménez 2012], Jimenez propose une revue des méthodes de planification basées sur les modèles continus, discrets et géométriques. Cette revue centrée sur les modèles est mise à jour et étendue dans [Arriola-Rios 2020], les auteurs y présentent en détail les modèles les plus récents en incluant les modèles basés sur l'apprentissage et les données. Les auteurs, dans [Herguedas 2019], se concentrent eux sur la manipulation à plusieurs bras.

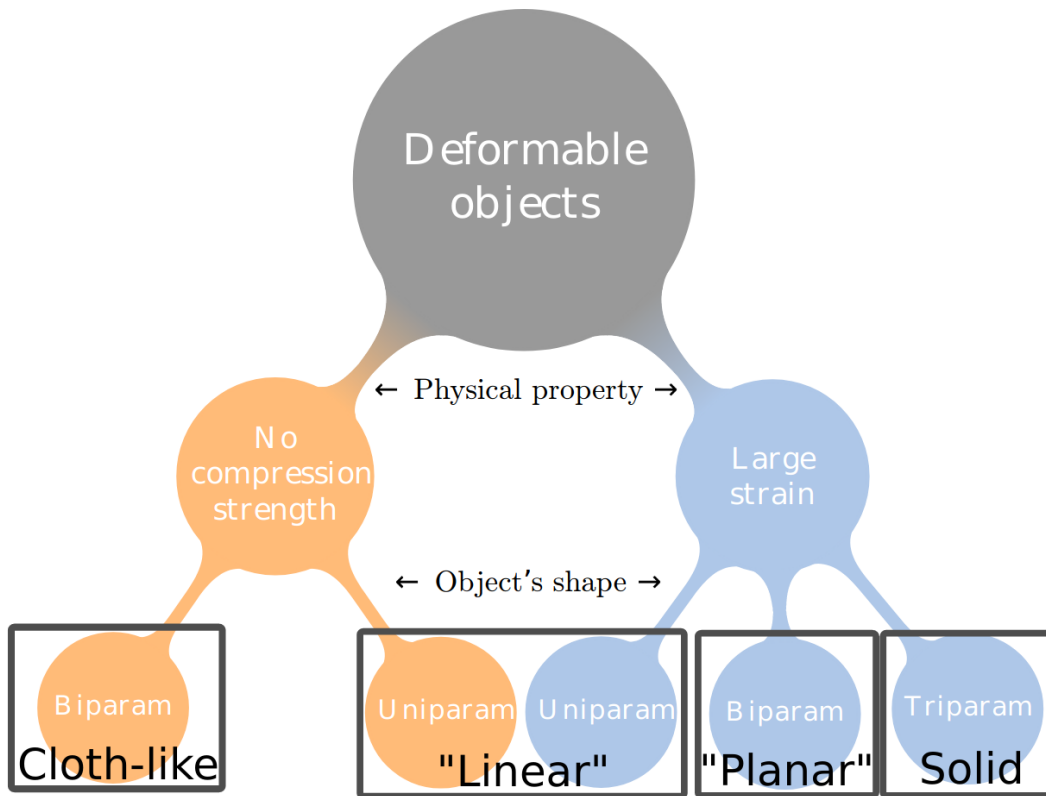


FIGURE 2.1 – Classification des objets déformables dans [Sanchez 2018]

A l'inverse des revues précédentes qui classifient les modèles par type, Sanchez et al., dans [Sanchez 2018] classifient les objets en fonction de leur géométrie et leurs propriétés physiques (voir figure 2.1). Les auteurs proposent de répertorier l'ensemble des objets déformables dans quatre catégories (figure 2.1) :

- **Textiles** : Les objets plans ne possédant que très peu de forces de compression interne et ayant une dimension très petite devant les deux autres.
- **Linéaires** : Les objets ayant une dimension très grande devant les deux autres. Les auteurs distinguent deux sous-catégories : les objets linéaires sans force de compression comme les câbles souples et les cordes et les objets avec une tension interne comme les poutres et les câbles rigides.
- **Planaires** : L'ensemble des objets plans avec des tensions internes non négligeables. Inclue également les objets creux tels que les balles de caoutchouc vides ou les bouteilles en plastique.
- **Solides** : L'ensemble des objets 3D avec des tensions internes. Ce sont généralement les plus difficiles à modéliser. Contrairement aux

2.1. Caractérisation et classification des méthodes de modélisation

autres types, on ne peut pas réduire la taille du problème en négligeant les déformations selon certaines directions. Cette catégorie inclue l'ensemble des objets volumineux non creux, tels que les fruits et la nourriture en général.

Cette classification permet de mettre en avant, par rapport au type de tâche que l'on cherche à réaliser, les points forts des différents modèles et leurs domaines d'application. Les objets déformables sont un ensemble extrêmement vaste et très hétérogène qui regroupe un ensemble d'objets aux propriétés différentes. Il est donc naturel que l'ensemble des méthodes utilisées pour les manipuler reflètent ces caractéristiques. Cette classification permet donc d'avoir une vue d'ensemble et du contexte dans lesquels chaque approche est appliquée. En effet, la modélisation et la stratégie de contrôle sont choisies en fonction du type d'objet, des particularités de la tâche, et des outils à disposition (données, matériel, perception, textures, etc).

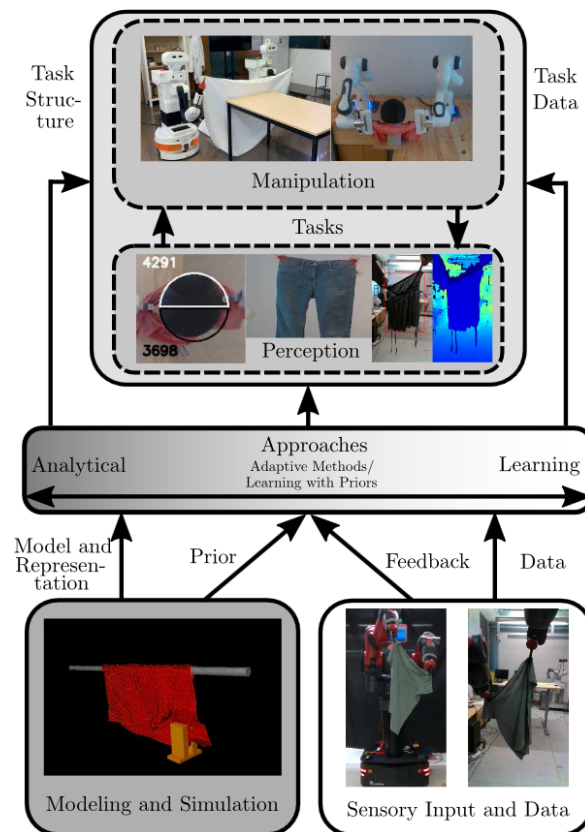


FIGURE 2.2 – Classification proposée dans [Yin 2021]. Continuum des approches analytiques aux approches basées données

Plus récemment, les auteurs de [Yin 2021] adressent l'utilisation et l'intégration de modèles classiques en conjonction avec des approches basées

données pour résoudre les problèmes ouverts pour la perception et la manipulation. Ils dégagent ainsi les principales pistes de recherche ainsi que les développements théoriques nécessaires d’un point de vue apprentissage. Leur classification est exposée sur la figure 2.2 où les approches sont présentées sur un continuum analytique - apprentissage. De manière plus large, la revue [Zhu 2022] se concentre sur l’identification des principaux problèmes ouverts liés à chaque aspect de la manipulation robotique d’objet déformable. Ils mettent en avant les challenges sur les préhenseurs, le design de robot, mais également la perception, la planification, le contrôle et la modélisation. Une représentation du cadre robotique incluant ces éléments est donnée sur la figure 1.1.

Dans ce chapitre, nous allons répertorier les différentes approches en fonction des types de modèles utilisés. On peut classifier ces modèles en quatre catégories distinctes :

- **Modèles physiquement réalistes**
- **Modèles discrets**
- **Approches basées données**
- **Asservissement visuel**

Nous allons présenter leurs avantages et inconvénients dans le cadre de cette thèse et en prenant en compte les objectifs fixés pour ainsi justifier le choix du modèle Cosserat.

2.1.1 Modèles discrets

2.1.1.1 Modèle Masse-ressort-amortisseur (MSD)

Afin de modéliser les déformations, les objets sont modélisés comme un ensemble discret de points. Les modèles Masses-ressorts-amortisseurs (MSD - Mass-spring-damper) représentent ces points comme des masses discrètes. Chaque masse est reliée à ses voisins par un ensemble de ressorts et d’amortisseurs virtuels (figure 2.4). Les efforts internes sur chaque masse sont calculés à partir des forces visco-élastiques en fonction des déplacements et des vitesses des masses. Ces contraintes sont ensuite linéarisées autour de l’état actuel pour résoudre le système avec la méthode de Newton [Müller 2008] par exemple.

Les avantages des MSD sont leur faible coût en calcul et leur facilité d’implémentation. Ils sont également faciles à généraliser. La modularité de la structure des ressorts et amortisseurs, en jouant sur leur nombre et agencement (figure 2.4), permet de modéliser de nombreux types d’objet. Ils ont été utilisés dans [Wada 1998] pour modéliser des objets plans, dans [Hirai 2001, Leizea 2015] pour des éponges et objets solides, et dans

2.1. Caractérisation et classification des méthodes de modélisation

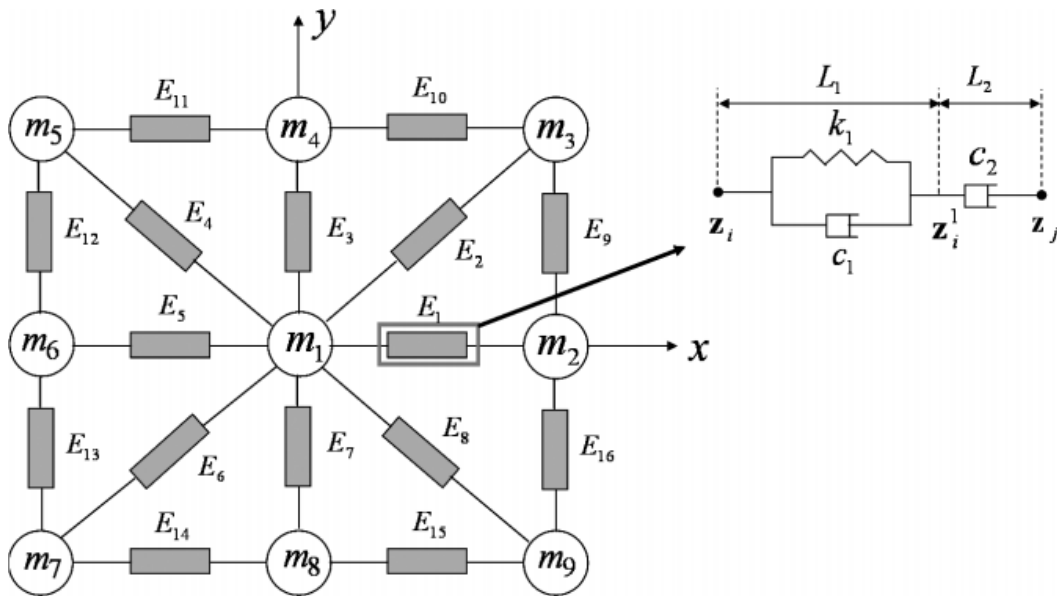


FIGURE 2.3 – Exemple de maillage MSD, avec la configuration des ressorts et amortisseurs utilisée dans [Das 2011]

[Wang 2010b, Sakamoto 2009] pour modéliser de la nourriture pour un processus industriel. Le modèle permet notamment de traiter les déformations rhéologiques, c'est-à-dire les déformations qui ont une composante plastique, en adaptant les paramètres et la structure des ressorts et amortisseurs aux comportements rhéologiques [Ikawa 2007, Cocuzza 2018].

La méthode est toutefois peu adaptée à la modélisation de déformations larges et complexes. De plus, augmenter la résolution du maillage ne garantit pas la convergence vers la solution désirée, contrairement à d'autres modèles. Enfin, il est difficile de paramétrer le système de sorte qu'il reflète la réalité en termes d'efforts. Le nombre de masses, ainsi que les constantes de raideur des ressorts n'ont pas de sens physique associé. Dans [Wang 2009, Wang 2010a], les auteurs estiment ces constantes en calibrant leur modèle sur un modèle basé sur la méthode des éléments finis (FEM).

2.1.1.2 Systèmes de particules

Contrairement aux modèles MSD, ces approches n'utilisent pas de maillages. L'objet est modélisé comme un système discret de particules soumis à des contraintes holonomes. Ils sont généralement appelés Position-Based Dynamics (PBD). Lorsque l'objet est soumis à une force extérieure, le déplacement de chaque particule est calculé en fonction du déplacement des particules adjacentes et des paramètres des contraintes appliquées sur le système. Les forces internes sont obtenues implicitement à partir de ces

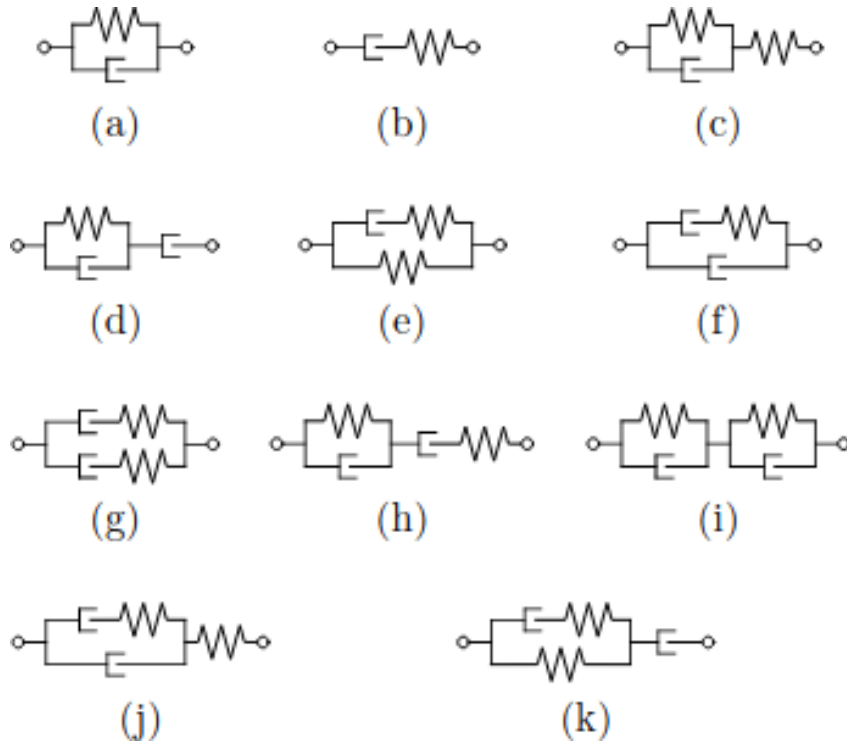


FIGURE 2.4 – Différentes configurations de ressorts-amortisseurs reliant les masses entre elles [Tokumoto 2001]

paramètres et des efforts extérieurs appliqués [Bender 2015]. Ces paramètres gouvernent donc le comportement élastique de l'objet. Les avantages de ces modèles sont similaires aux MSD. En plus d'être généralement plus stables que ces derniers, ils sont également rapides à calculer [Macklin 2014, Macklin 2016]. Ils peuvent être adaptés à de nombreux objets et problèmes différents en rajoutant des contraintes aux limites.

Toutefois, leur utilisation reste majoritairement cantonnée aux domaines de l'animation graphique et de la perception [Guler 2015]. En effet, bien que les déformations obtenues soient plausibles visuellement, les mesures de force sont peu précises et doivent être récupérées à partir d'une formulation implicite. Le modèle est donc difficile à utiliser dans une stratégie de contrôle. Dans ce cas, les efforts appliqués par les actionneurs pour déformer un objet doivent être déterminés de manière fidèle. De plus, les paramètres qui gouvernent le comportement de l'objet n'ont pas de sens physique évident et sont difficiles à estimer. Güler et al. dans [Güler 2017] ont intégré un paramètre de déformabilité dans un modèle Meshless Shape Matching (MSM) qui est un type de PBD. Ce paramètre, de manière similaire aux MSD, est calibré en comparant le comportement de leur simulation par rapport à un modèle FEM. Ce dernier sert de vérité terrain.

2.1. Caractérisation et classification des méthodes de modélisation

Un autre modèle bien connu de la communauté d'animation graphique est le modèle As-Rigid-As-Possible (ARAP) [Sorkine 2007, Levi 2014, Dou 2017, Parashar 2015]. Le modèle ARAP permet de modéliser géométriquement un objet en partant de l'hypothèse que ce dernier préserve sa rigidité localement. L'objet est modélisé par une mesure de l'énergie représentant la somme des déviations locales par rapport à l'équilibre sur la surface de l'objet [Terzopoulos 1987]. Ce dernier a récemment été appliqué à la manipulation robotique d'objets déformables. Shetab et al. dans [Shetab-Bushehri 2022], utilisent le modèle ARAP d'une part pour détecter un objet déformable, et d'autre part pour déterminer une matrice Jacobienne. Cette Jacobienne relie la position et l'orientation des préhenseurs de deux robots aux déplacements de points de contrôle du modèle. La comparaison entre la forme désirée de l'objet et la forme obtenue est représentée sur la figure 2.5.

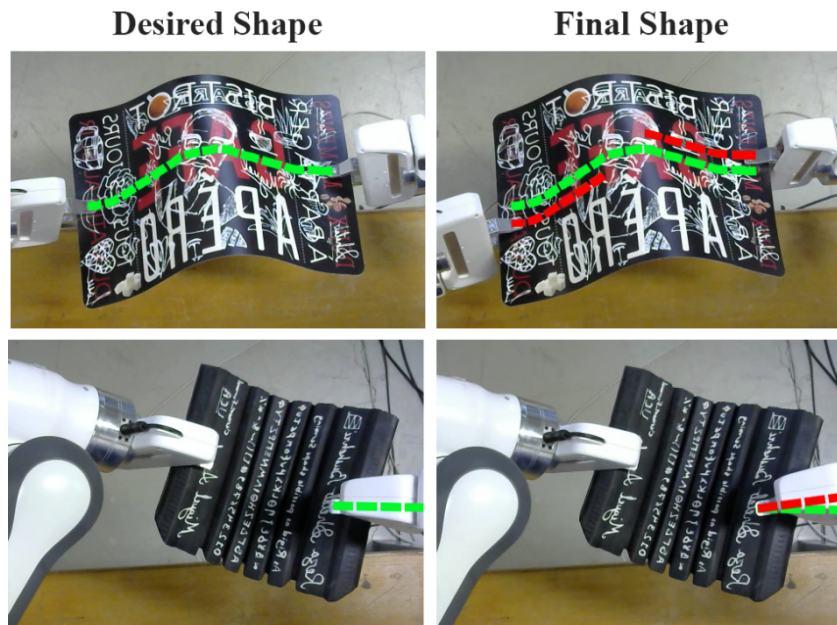


FIGURE 2.5 – Comparaison entre la forme obtenue et la forme désirée de l'objet dans [Shetab-Bushehri 2022]

2.1.2 Modèles physiquement réalistes

2.1.2.1 Modèle volumique et surfacique

Récemment, les modèles physiquement réalistes se sont imposés comme une méthode centrale de modélisation d'objets déformables. Ils offrent la meilleure précision et permettent de modéliser physiquement et de manière continue les déformations d'un objet. Ils se basent sur la théorie pré-établie

de la mécanique des milieux continus. Ils sont généralement formulés sous la forme d'équations aux dérivées partielles (PDE). Le plus utilisé pour la modélisation d'objets déformables est la méthode des éléments finis (FEM). Cette méthode consiste à représenter un objet volumique par un maillage consistant en un nombre fini d'éléments géométriques. Pour modéliser un objet volumique, ces éléments sont souvent des tétraèdres, et des triangles pour un objet en deux dimensions. Afin de maintenir la consistance du maillage, des contraintes sont appliquées aux éléments aux limites. Au sein de chaque élément, une forme réduite des équations de comportement est convertie en équations différentielles ordinaires (ODE) qui sont résolues en les intégrant numériquement à partir des valeurs aux limites de l'élément. L'intégration requiert de résoudre le terme lié aux forces internes en linéarisant à chaque étape. L'approximation linéaire donne de bons résultats à l'exception du champ de rotation en cas de larges déformations. La méthode co-rotationnelle [Hsiao 1999] adresse ce problème en ajoutant un terme de compensation.

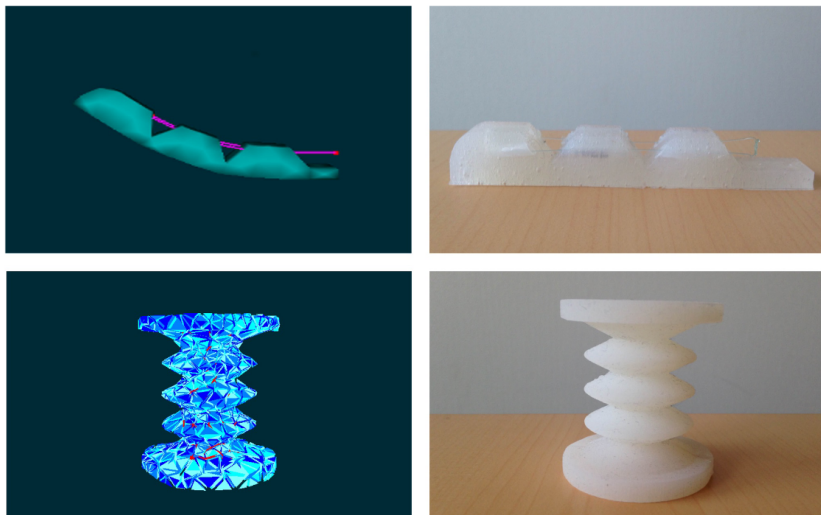


FIGURE 2.6 – Comparaison entre des actionneurs déformables et leur simulation dans SOFA [Duriez 2016]

La dimension de l'espace des configurations d'un modèle FEM peut être réduite en se concentrant sur les principaux modes de déformation linéaires [Qu 2004, Goury 2018]. Ce principe est étendu aux déformations non linéaires dans [Kerschen 2009, Sin 2013, Chenevier 2018]

L'efficacité de la résolution en temps réel du modèle FEM a permis l'émergence du simulateur SOFA, basé sur les éléments finis [Faure 2012]. Le cadre de SOFA a initialement été développé pour modéliser des interactions chirurgicales. Il a depuis été étendu à la planification pour la manipulation d'objets déformables [Yoshida 2015, Ficuciello 2018] et au

2.1. Caractérisation et classification des méthodes de modélisation

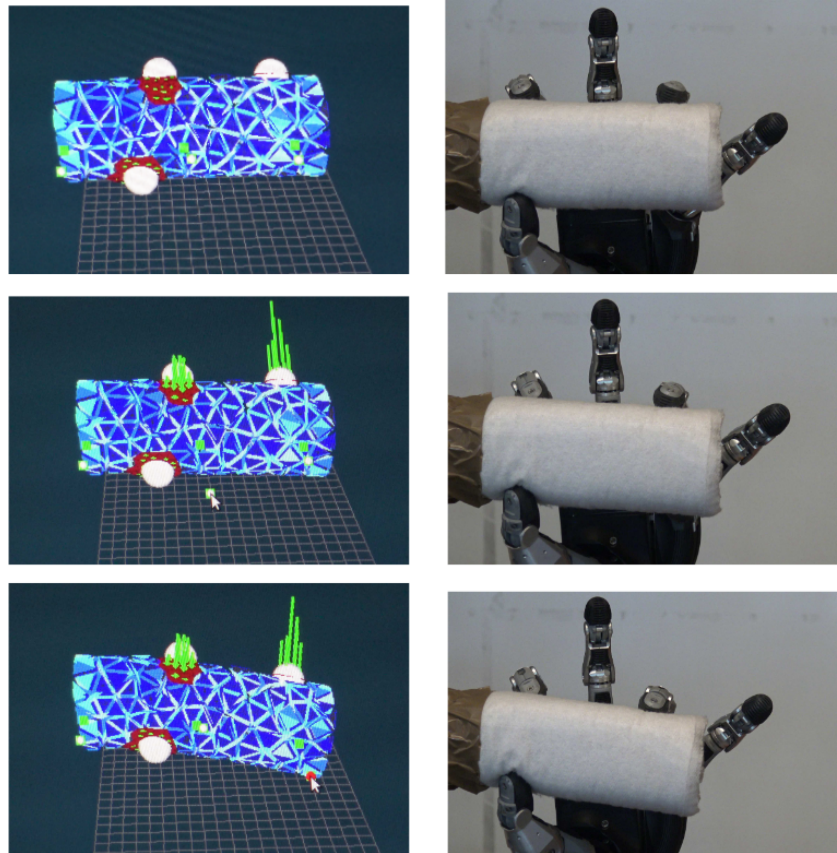


FIGURE 2.7 – Comparaison entre un objet déformable réel et l’objet simulé avec des éléments finis dans SOFA [Ficuciello 2018]

contrôle de robots déformables [Duriez 2013, Duriez 2016, Adagolodjo 2021, Coevoet 2017, Coevoet 2019]. Les figures 2.6 et 2.7 illustrent respectivement la comparaison entre des actionneurs déformables réels et simulés, et entre un objet manipulé en temps réel et sa simulation dans SOFA.

2.1.2.2 Modèle linéique

Bien que le modèle FEM puisse facilement être étendu aux objets linéaires [Grazioso 2019, Duenser 2018, Koessler 2021], des techniques de réduction de modèle sont nécessaires pour l’inclure dans une boucle de contrôle [Goury 2018]. Le modèle Cosserat a historiquement démontré d’excellentes performances pour la modélisation d’objets déformables linéaires et robots déformables [Chirikjian 1993, Macchelli 2009, Gravagne 2003]. Plus récemment, Campisano et al. ont intégré une formulation du modèle dans un contrôleur en boucle fermée afin de contrôler la position et l’orientation d’un robot déformable [Campisano 2021].

Dans le cas des robots déformables, le point d’actionnement est rarement situé à la base. Cela signifie que les actionneurs appliquent des efforts sur le bout du robot, non fixé en position. Dans le cas spécifique de la manipulation d’objets déformables, le robot contrôle la position et l’orientation de l’objet au point de contact, donc potentiellement à la base. Cela signifie que les équations du modèle cinématique direct de Cosserat peuvent être exprimées sous forme d’équations différentielles ordinaires (ODE). Résoudre un système ODE est alors bien plus facile que de résoudre un système PDE. Pour ces raisons, sa précision, son lien physique à la réalité ainsi que sa capacité à être implémenté dans un contrôleur en boucle fermée, c’est le modèle que nous avons choisi pour modéliser les objets déformables dans cette thèse. Nous reviendrons en détail sur la théorie de Cosserat dans la section suivante.

On peut noter qu’il existe aussi des formulations de la théorie de Cosserat pour modéliser les objets plans et volumiques. Cette formulation est également basée sur un ensemble d’éléments [Riahi 2009, Riahi 2010]. Celles-ci ne seront toutefois pas abordées dans cette thèse.

Ces méthodes ont néanmoins deux désavantages. Le premier est leur coût en calcul élevé, il est encore très difficile de les intégrer dans une boucle de contrôle pour un objet réel lorsque l’on tente de résoudre le modèle inverse. Le second est que leur performance est extrêmement sensible à l’estimation des paramètres élastiques de l’objet modélisé (module d’Young E et coefficient de poisson ν). L’estimation de ces coefficients est un domaine de recherche à part entière en robotique, autant pour le modèle Cosserat [Berthold 2021] que pour le modèle FEM.

C’est ce modèle que nous avons choisi d’utiliser dans cette thèse, d’une part pour son réalisme qui nous permet de modéliser les efforts élastiques dans l’objet, et d’autre part pour sa formulation sous forme d’ODE. Bien que le modèle inverse soit effectivement difficile à résoudre rapidement, il suffit en revanche d’intégrer spatialement les équations pour résoudre le problème aux valeurs initiales. De plus, le problème aux valeurs initiales est bien posé. On utilisera ainsi ces deux propriétés par la suite pour mettre en place notre contrôleur sans résoudre le coûteux problème inverse.

2.1.2.3 Estimation des constantes élastiques

En pratique, ces paramètres sont souvent ajustés manuellement en les modifiant petit à petit jusqu’à ce que les résultats de simulation semblent visuellement plausibles. Bien que cela puisse être applicable dans le cas de l’animation 3D, cela l’est beaucoup moins lorsque des objets réels, soumis à un ensemble de contraintes et susceptibles d’être endommagés sont manipulés.

2.1. Caractérisation et classification des méthodes de modélisation

L'approche privilégiée dans la littérature est d'appliquer des efforts sur l'objet dont on cherche à identifier les coefficients et de comparer la déformation obtenue avec celle prédite par le modèle. La différence entre les deux est alors minimisée avec comme variables d'optimisation les constantes élastiques. Dans [Frank 2010], les auteurs appliquent une pression sur différents objets et comparent les données des capteurs avec un modèle FEM de l'objet. Dans [Fugl 2012] les auteurs utilisent un modèle d'Euler-Bernoulli pour estimer les paramètres d'un objet se déformant sous l'action de la gravité. De manière similaire, Petit et al. dans [Petit 2017], puis dans [Petit 2018] utilisent un modèle FEM co-rotationnel ainsi que des capteurs tactiles et RGB-D afin d'estimer ces paramètres.

2.1.3 Approches basées données

Afin d'adresser la dimension importante de l'espace d'état des objets déformables, et les difficultés de modélisation que cela cause, les approches basées données ont émergé récemment pour résoudre des tâches de manipulation d'objets déformables.

2.1.3.1 Modèles appris

Ces approches utilisent des données d'apprentissage pour entraîner des réseaux de neurones à apprendre un modèle dynamique des déformations d'un objet [Cretu 2011, Tanaka 2018, Valencia 2020]. Elles sont formulées comme un problème d'apprentissage supervisé. Ces modèles appris peuvent inclure des modèles physiques [Battaglia 2016] et ainsi modéliser des déformations non-linéaires avec un bon degré de précision, et de manière rapide. Dans [Holden 2019], les auteurs apprennent un modèle pour simuler interactivement des textiles. Yan et al. dans [Yan 2020] se basent sur une représentation d'état du système physique pour contrôler la forme de cordes souples. Le modèle inverse peut également être appris. Les auteurs de [Lippi 2020] se basent sur la vision et sur un réseau de neurones pour prédire les actions d'un robot pour une tâche de manipulation de textiles. Ces approches sont toutefois très spécifiques aux objets dont le modèle a été appris et nécessitent un ensemble important de données d'apprentissage qui peuvent être difficiles à obtenir.

2.1.3.2 Apprentissage par démonstration

Les approches basées données présentées précédemment ont pour but d'apprendre un modèle. Ces approches en revanche, sont utilisées pour apprendre une tâche spécifique. Des exemples de tâches apprises sont : faire un noeud avec une corde [Kudoh 2015, Lee 2015], plier des textiles [Lin 2021,

[Schulman 2016] ou encore faire une suture [Schulman 2013]. Les données d'entraînement sont typiquement obtenues à partir d'un expert humain, par exemple en téléopération [Rambow 2012]. Les primitives de mouvement sont extraites des données et généralisées pour réaliser la tâche souhaitée. Une difficulté spécifique à ces approches est de généraliser les solutions obtenues à partir d'un faible nombre de données [Pignat 2017, Schulman 2016].

2.1.3.3 Apprentissage par renforcement

Afin d'outrepasser le problème des données d'apprentissage, et grâce au développement récent de simulateurs efficaces pour modéliser les objets déformables, des approches se basent sur de l'apprentissage par renforcement pour générer des politiques de contrôle en simulation. La procédure d'apprentissage est conduite lors de multiples essais pour maximiser une récompense. Dans de nombreux cas, les récompenses sont rares, une exploration complète est donc rarement utilisée en pratique. La connaissance du domaine, de la tâche, et le choix de la politique d'initialisation sont donc cruciaux pour générer une politique de contrôle applicable à un robot réel. Utiliser une politique d'initialisation provenant de démonstrations expertes a prouvé son efficacité en générant de la diversité en quelques itérations à peine [Balaguer 2011, Tsurumine 2019, Twardon 2018].

Il est donc possible de générer des politiques qui fonctionnent en simulation. Toutefois cela demande un travail d'ingénierie conséquent pour ajouter manuellement des sous-tâches et définir la fonction récompense qu'échangent les agents lors de l'apprentissage. De plus, bien que l'efficacité des simulateurs pour les objets déformables ait beaucoup augmenté ces dernières années, ils n'ont pas encore un degré de stabilité comparable avec les simulateurs d'objets rigides, notamment au niveau de la gestion des contacts. La conséquence est que l'algorithme peut potentiellement exploiter des défauts du simulateur et générer des trajectoires qui ne sont pas physiquement possibles. De plus, lorsqu'elles sont déployées sur un système réel, les politiques apprises en simulation doivent faire face à un important écart entre la réalité et la simulation. Ce "sim-to-real gap" est particulièrement important lorsque des objets déformables sont impliqués et le taux de réussite des politiques transférées est souvent diminué de moitié [Matas 2018, Clegg 2018]. Ce problème est accentué par le fait que, dans leur travaux, Matas et al. et Clegg et al. utilisent MuJoCo [Mordatch 2012, Brockman 2016] et Bullet pour apprendre leur politique. Bien que ces deux simulateurs soient parfaitement adaptés à l'apprentissage en simulation, ils utilisent un dérivé des modèles PBD pour simuler les objets déformables. Or, comme illustré dans les sections précédentes, ces types de modèles ne produisent pas des déformations aussi

réalistes que les modèles physiques comme les FEM. Cela explique en partie pourquoi ces méthodes ont été appliquées à la manipulation d'objets textiles, qui fonctionne mieux dans ces simulateurs.

En conclusion, les modèles basés données offrent des perspectives intéressantes et sont déjà efficaces, notamment pour la manipulation de textiles. L'obtention des données d'apprentissage reste toutefois encore un problème ouvert. De plus, les tâches et modèles appris sont très spécifiques et difficiles à généraliser.

2.2 Modèle Cosserat

2.2.1 Applications robotiques - cinématique et dynamique

Le modèle Cosserat est l'un des modèles physiquement réalistes les plus précis pour modéliser les objets linéaires. Les déformations volumiques de l'objet modélisé sont négligées en se concentrant uniquement sur les déformations de l'axe central. Cette hypothèse permet de limiter la dimension du problème statique en le formulant sous la forme d'un ensemble d'équations différentielles ordinaires (ODE), ou le problème dynamique sous la forme d'équations aux dérivées partielles (PDE). Ce modèle mécanique a principalement été utilisé en robotique pour modéliser des robots déformables linéaires. Ces robots ont la particularité d'avoir une dimension physique très grande devant les deux autres. Les hypothèses du modèle sont vérifiées en pratique et se prêtent parfaitement à ce type de géométrie. Cela permet de décrire l'objet de manière précise en se reposant sur le cadre pré-établi de la mécanique continue des poutres.

Les auteurs dans [Trivedi 2008, Rucker 2010, Dupont 2009, Jones 2009] ont utilisé un modèle géométrique exact de Cosserat couplé à une formulation statique du problème pour contrôler un manipulateur souple formé de tubes concentriques. Ce cadre de travail a été étendu à une formulation dynamique en ajoutant le temps dans les équations [Lang 2011, Rucker 2011, Renda 2014, Alqumsan 2019]. Les termes dynamiques ainsi que les équations de compatibilité changent alors le système ODE en un système d'équations paraboliques aux dérivées partielles. Cela implique d'utiliser des schémas d'intégration plus complexes pour résoudre le problème, comme le schéma de Lax-Wendroff qui est d'ordre deux en espace et en temps [Lax 1958]. L'augmentation du coût computationnel est ensuite amplifiée en résolvant le problème aux valeurs limites avec les approches classiques comme la méthode de tir [Rao 2009]. Ces approches impliquent d'intégrer les équations un grand

nombre de fois afin de trouver une solution qui respecte les conditions aux limites.

Ces formulations ont été utilisées pour contrôler des robots déformables linéaires à tubes concentriques (voir figure 2.8), actionnés par des tendons [Rucker 2013, Starke 2017, Oliver-Butler 2019, Amanov 2021, Janabi-Sharifi 2021], des composants magnétiques [Kratchman 2016, Edelmann 2017], des injections de fluides [Renda 2020], ou encore des robots parallèles [Bryson 2014, Black 2017]. Dans [Adagolodjo 2021], les auteurs couplent le modèle Cosserat au modèle à éléments finis pour simuler un robot déformable. Les parties volumiques du manipulateur sont modélisées avec un modèle FEM tandis que les tendons qui servent d’actionneurs sont modélisés avec un modèle Cosserat.



FIGURE 2.8 – Robot à tubes concentriques [Rucker 2010]

En parallèle, le modèle a également été utilisé, principalement dans le domaine de l’animation et de la simulation, pour simuler le comportement dynamique d’objets déformables linéaires tels que des cordes ou filaments [Spillmann 2007, Lang 2011, Gazzola 2018, Orekhov 2020], voir figure 2.9.

Bien que la théorie de Cosserat puisse être adaptée directement pour calculer la cinématique directe d’un objet, l’inversion du modèle statique est beaucoup plus complexe. Dans [Rucker 2013, Fallahi 2016, Khadem 2016], les auteurs ont montré, même sans retour sur la pose de l’objet, l’efficacité du calcul en temps réel de la cinématique directe. Ils utilisent une matrice Jacobienne qui crée un lien entre la vitesse du robot, et la vitesse donnée aux actionneurs. Dans une série de travaux [Till 2015, Till 2017a, Till 2019], les auteurs proposent de résoudre le problème dynamique direct en temps réel. Leur méthode permet de réduire les PDE à des ODE en discrétisant

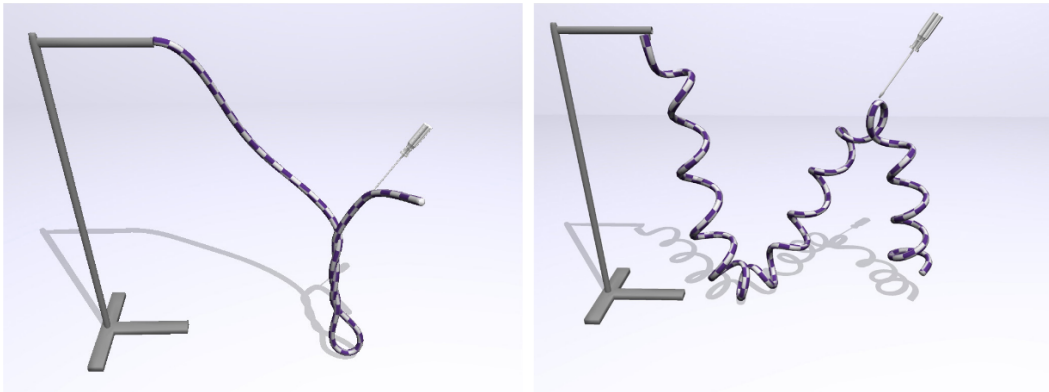


FIGURE 2.9 – Simulation interactive d'objets linéaires déformables [Spillmann 2007]

implicitement le temps. Le problème aux valeurs limites qui en résulte est ainsi résolu pour obtenir une solution au problème cinématique direct à chaque itération, pour une application en simulation. La boucle en simulation est illustrée sur la figure 2.11.

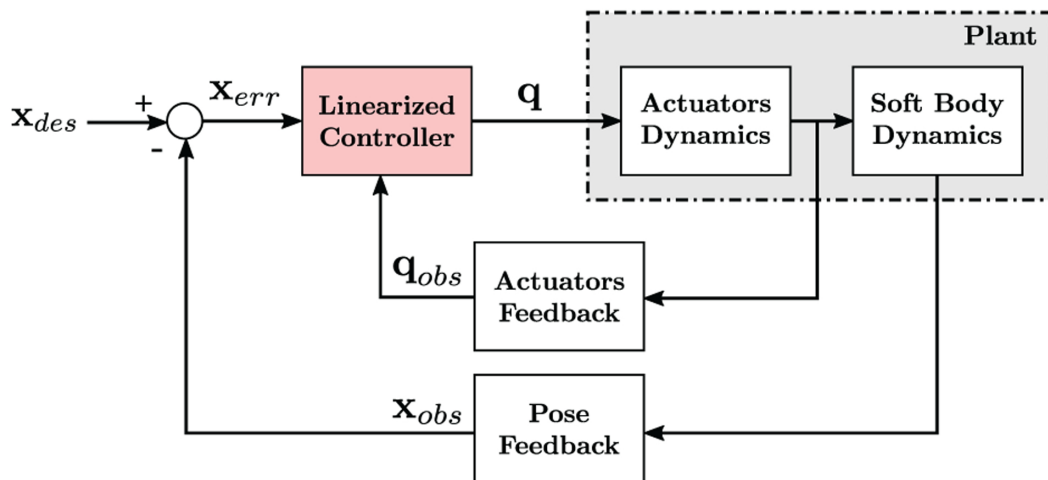


FIGURE 2.10 – Boucle de contrôle avec retours sur la pose et les actionneurs [Campisano 2021]

Toutefois, de nombreuses hypothèses simplificatrices sont nécessaires (hypothèses sur les matériaux, estimations des paramètres élastiques, pré-courbure initiale). Ces hypothèses ne sont pas toujours remplies et le modèle Cosserat, comme tous les modèles physiquement réalistes est extrêmement sensible à ces paramètres. En l'absence de retour sur la pose courante, il en résulte des erreurs qui peuvent être relativement importantes si ces paramètres ne sont pas estimés très précisément. De plus, ces erreurs

peuvent être amplifiées lorsque l'on essaie d'atteindre des formes complexes de l'objet. Ces dernières excitent les modes de déformation contenant le plus de non-linéarités. Afin d'éviter ces erreurs statiques, les auteurs dans [Campisano 2021] ajoutent un retour à la fois sur les actionneurs et sur la pose de l'objet (voir figure 2.10). Ces retours sont utilisés pour calculer une matrice Jacobienne qui associe les couples appliqués par les actionneurs à la pose du bout du robot, permettant ainsi de contrôler en boucle fermée sa position et son orientation. On appelle le bout, le point extremum de la tige constituant le robot. Ce point se situe à la limite de l'objet et dans la plupart des applications de contrôle de robot linéaire déformable, c'est ce point ci que l'on cherche à contrôler.

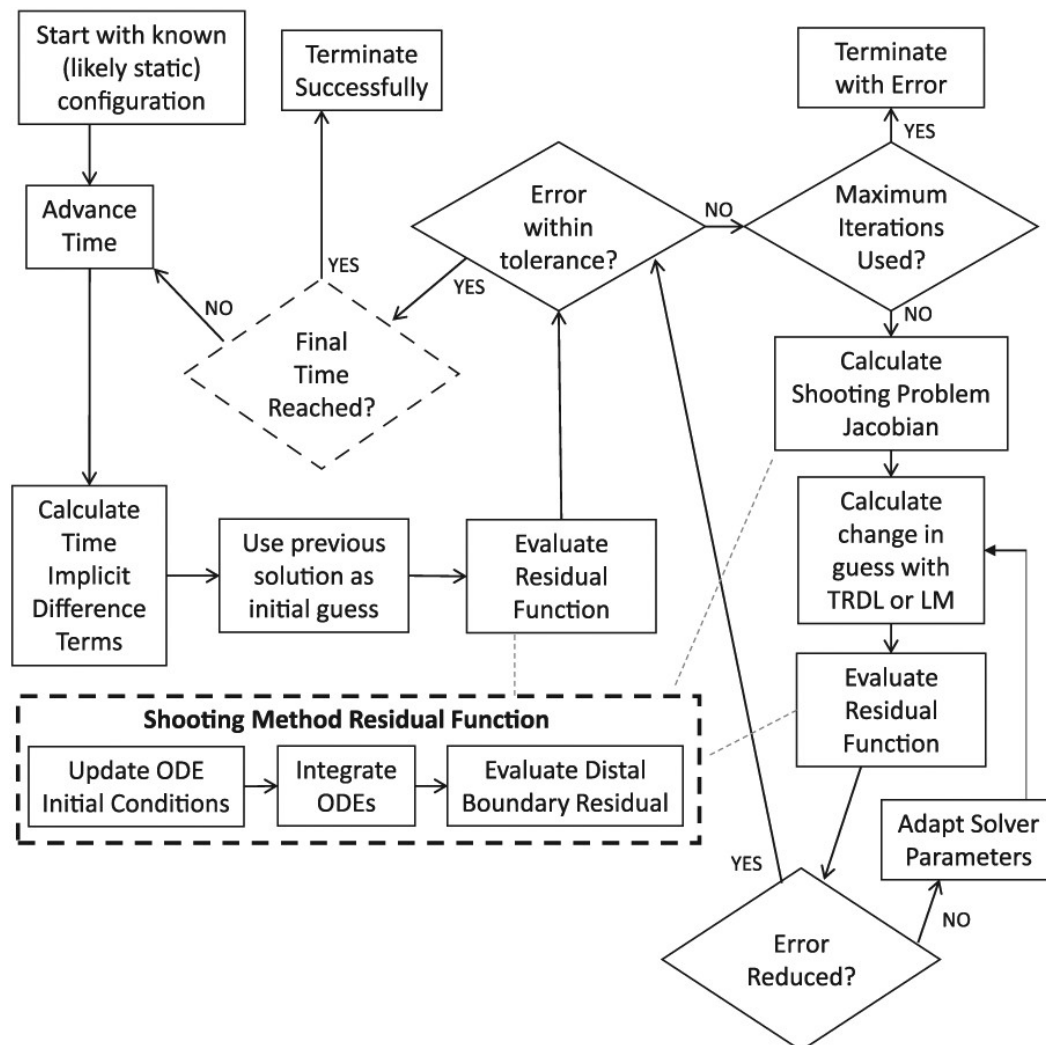


FIGURE 2.11 – Boucle de simulation en temps réel, la partie gauche correspond à la boucle temporelle de la simulation [Till 2019]

2.2.2 Solution optimale

Le problème direct cinématique et dynamique de Cosserat permet de déterminer la forme de l'objet modélisé résultant de la commande des actionneurs. Résoudre le problème inverse permet quant à lui, de trouver les conditions initiales qui, intégrées, donnent une solution qui respecte un ensemble de conditions (valeurs limites, conditions limites, position et/ou orientation de points de l'objet). Pour la plupart des applications de contrôle, c'est ce problème qu'il va être intéressant de résoudre. Résoudre le problème inverse permet par exemple de déterminer les commandes des actionneurs pour contrôler le bout d'un robot déformable [Rucker 2011], ou encore les commandes de manipulateurs pour déformer l'objet dans une pose souhaitée. En revanche, pour simuler un objet ou robot, on va au contraire essayer de résoudre le modèle direct de sorte à obtenir la pose de l'objet soumis à la commande transmise par les actionneurs.

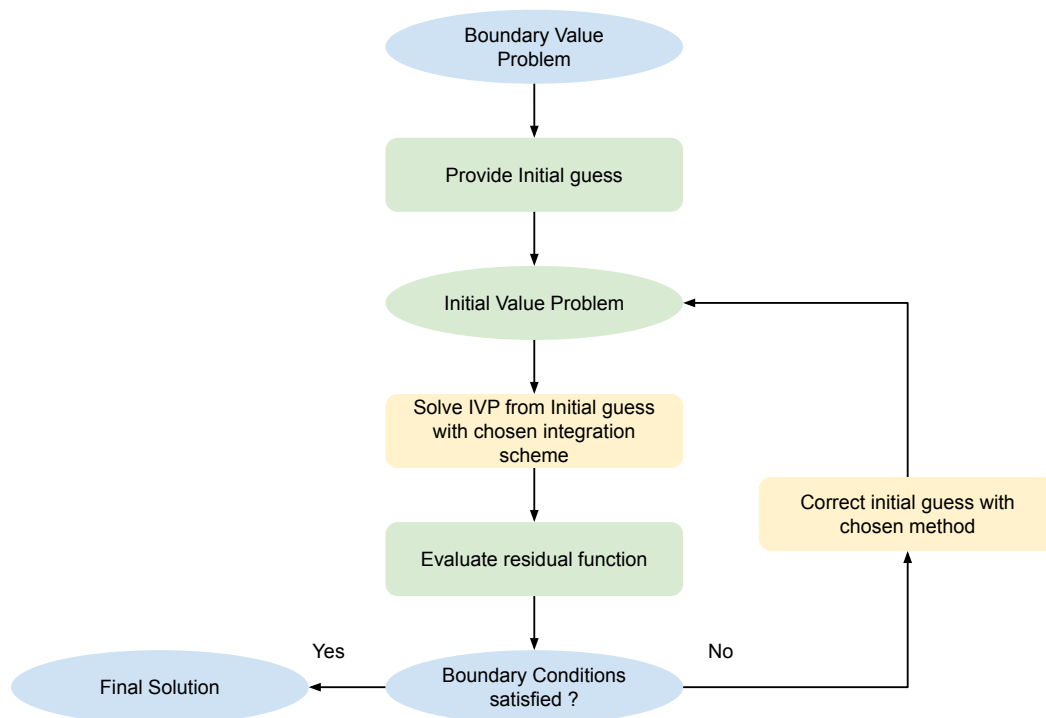


FIGURE 2.12 – Méthode de tir - schéma

A l'exception des approches Lagrangiennes, comme les éléments finis géométriquement exactes [Simo 1985], les tiges discrètes [Bergou 2008] et les paramétrisations basées efforts [Boyer 2020], la vaste majorité des modèles inverses et dynamiques sont formulés comme des problèmes aux conditions limites (BVP - boundary value problem). Une solution analytique

étant difficile à trouver, ces problèmes sont résolus numériquement avec des approches classiques de résolution de BVP comme la méthode de tir [Rao 2009]. Cette méthode permet de résoudre un BVP en résolvant successivement des problèmes aux valeurs initiales (IVP - initial value problem). Les conditions initiales sont intégrées, évaluées, puis corrigées jusqu'à trouver une solution intégrée qui respecte les conditions aux limites imposées dans le BVP (voir figure 2.12).

Il est important de noter que le problème direct peut également être formulé comme un problème aux valeurs initiales (IVP - initial value problem). Ce problème est résolu en intégrant les équations de Cosserat à partir des valeurs initiales. Pour une formulation statique, cela revient à intégrer spatialement les équations le long de la tige, robot ou objet. Le problème étant bien posé [Trivedi 2008], la solution obtenue est unique. Les problèmes qui peuvent être résolus uniquement en intégrant les équations de Cosserat sont toutefois peu communs. Il est en effet extrêmement rare de connaître précisément l'ensemble des valeurs initiales à la base de la tige.

Dans la plupart des travaux illustrés, les BVPs formulés sont déduits des équations cinématiques de la tige et des lois de Newton. L'application de la théorie de la commande optimale au problème Cosserat statique est bien connue [Jurdjevic 1995, Jurdjevic 1997]. Elle a été appliquée pour étudier la stabilité de robots concentriques [Ha 2015, Ha 2018] et parallèles [Till 2017b].

Afin de modéliser la dynamique et la statique de robots déformables continus, les auteurs dans [Boyer 2022], formulent leurs BVPs à partir de la théorie de la commande optimale. Ils reformulent le problème dynamique direct comme un problème de commande optimale et montrent que le BVP obtenu contient naturellement un modèle continu de l'accélération. Le problème d'optimisation est formulé de sorte à trouver les commandes optimales, efforts en statique et accélérations en dynamique, qui minimisent l'énergie potentielle élastique dans le robot. Les contraintes dépendent également du type de modèle, statique ou dynamique. L'accélération, bien qu'elle joue un rôle non-négligeable en simulation est souvent mise de côté au profit d'hypothèses simplifiant le problème.

2.2.3 Formulation Lagrangienne vs Newtonienne

Comme mentionné dans la section précédente, la plupart des travaux déduisent le BVP à partir des lois de mouvement de Newton. Les actionneurs sont inclus dans le modèle en tant que forces et moments extérieurs appliqués en certains points de l'objet. Le BVP résultant est alors résolu numériquement avec la méthode de tir. Cette approche a toutefois des limites : afin de réduire les coûts computationnels, les approches mentionnées utilisent des algorithmes

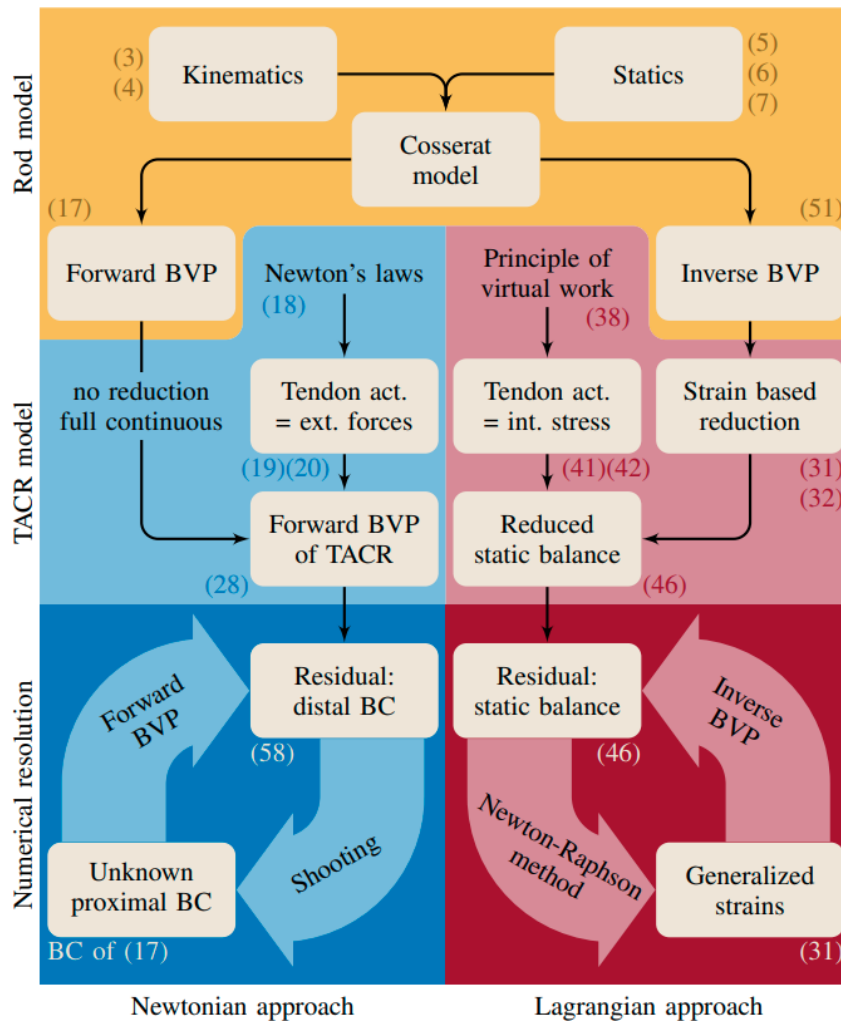


FIGURE 2.13 – Approches Newtonienne et Lagrangienne pour un robot déformable linéaire actionné par des tendons [Tummers 2023]

de recherche locale (LM, SQP, Interior point, trust region) et déterministes afin de mettre à jour les variables d'initialisation (initial guess sur la figure 2.12). Contrairement aux algorithmes stochastiques tels que les méthodes génétiques qui effectuent une recherche globale de la solution, les algorithmes locaux sont susceptibles de tomber dans des minima locaux. La convergence vers la solution globale dépend donc fortement des variables d'initialisation données à l'algorithme. De plus, pour des systèmes avec des structures complexes [Wang 2021], la dimension du vecteur d'état peut rapidement devenir très élevée.

Comme alternative à la méthode de tir, une formulation basée sur les équations de Lagrange a été développée [Renda 2020, Renda 2021, Boyer 2020]. Cette formulation permet de paramétrer la configuration de

l'objet par rapport aux efforts appliqués. Ce qui réduit la taille des équations, en particulier pour des systèmes complexes comportant de multiples actionneurs. Cette réduction de la taille du problème permet d'obtenir une formulation variationnelle [Zienkiewicz 2005] en linéarisant les équations d'équilibre et ainsi d'augmenter l'efficacité de la résolution à la fois en statique et en dynamique. Cette linéarisation prend généralement la forme d'une Jacobienne qui représente la variation de la configuration de l'objet par rapport à la variation des efforts soumis.

Dans [Tummers 2023], les auteurs comparent les deux approches de résolution et montrent qu'elles sont équivalentes pour un robot continu actionné par des tendons 2.13. On peut toutefois noter que la méthode de résolution, Lagrangienne ou méthode de tir, ne dépend pas nécessairement de la formulation du problème. Dans [Boyer 2022], les auteurs utilisent la méthode de tir pour résoudre le problème formulé du point de vue Lagrangien. Inversement, dans [Campisano 2021], les auteurs linéarisent les effets des actionneurs pour résoudre le problème formulé avec une approche Newtonienne.

2.3 Cadre mathématique du modèle Cosserat

On présente dans cette section le cadre mathématique du modèle Cosserat utilisé dans cette thèse. Le but de ce chapitre est de déterminer les équations différentielles non linéaires qui décrivent le comportement de poutres, tiges, et autres objets linéaires élastiques soumis à de grands déplacements. Les équations exactes pour de grands déplacements planaires d'objets linéaires ont été mises en formes par Euler en 1744, puis par Lagrange en 1762 pour des déplacements en trois dimensions. Ce modèle est exact dans le sens où il ne repose pas sur des approximations par rapport aux variables cinématiques [Antman 1976, Takens 1981] et qu'il propose une formulation exacte du tenseur de déformation. Les méthodes non-exactes utilisent typiquement deux approximations : (1) La forme déformée est supposée proche de la forme initiale lorsque les efforts et moments sont calculés, et (2) Une formule approximative est utilisée pour calculer la courbure de la tige. Nous n'utiliserons ainsi aucune de ces approximations dans la suite de cette thèse.

On utilise la formulation Newtonienne classique du problème reprise par Antman dans les chapitres 2 et 4 de [Antman 2005]. Cette formulation est à nouveau reprise par la communauté robotique, notamment dans [Trivedi 2008, Rucker 2010] avec des notations plus familières aux roboticiens [Murray 1994].

On utilisera le terme tige dans cette section pour se référer à la dimension principale d'un objet déformable linéaire modélisé par la théorie Cosserat.

2.3.1 Cinématique

2.3.1.1 Représentation géométrique

On définit la forme de la tige comme une courbe paramétrée dans l'espace Cartésien $p(s) \in \mathbb{R}^3$, couplée à une matrice orthogonale de rotation $R(s) \in SO(3)$ où $SO(3) = \{R \in \mathbb{R}_{3 \times 3} \mid R^T R = I, \text{ et } \det(R) = 1\}$. On identifie chaque point matériel par sa coordonnée s . La position et l'orientation de la tige sont des fonctions du paramètre s sur un intervalle fini correspondant à la longueur du Cosserat. On peut ainsi décrire la tige entière en associant s à la transformation rigide homogène $g(s) \in SE(3)$:

$$g(s) = \begin{bmatrix} R(s) & p(s) \\ 0^T & 1 \end{bmatrix} \quad (2.1)$$

Où $SE(3) = \left\{ \begin{bmatrix} R & b \\ 0 & 1 \end{bmatrix} \mid R \in SO(3), \text{ et } b \in \mathbb{R}_3 \right\}$. On s'intéresse à l'évolution de la forme de la tige par rapport à un état initial de référence qu'on note :

$$g^*(s) = \begin{bmatrix} R^*(s) & p^*(s) \\ 0^T & 1 \end{bmatrix} \quad (2.2)$$

Le symbole $*$ correspond ainsi aux variables associées à l'état de référence (voir fig 2.14). La courbure de référence, autrement dit la pré-courbure de la tige définit $p^*(s)$, mais l'orientation de référence le long de la courbe $R^*(s)$ peut être définie arbitrairement. On définit l'orientation de référence de sorte à ce que l'axe z soit orienté le long de la tangente à la courbe de référence :

$$R^*(s)\mathbf{e}_3 = \frac{\dot{p}^*(s)}{\|\dot{p}^*(s)\|} \quad (2.3)$$

Où $\dot{p}^*(s) = \frac{dp^*(s)}{ds}$, le \cdot représentant la dérivée spatiale par rapport à s . De plus, e_1, e_2 et e_3 sont utilisés pour désigner la base standard $[1, 0, 0]^T, [0, 1, 0]^T$ et $[0, 0, 1]^T$.

2.3.1.2 Géométrie différentielle

En suivant les notations définies dans [Murray 1994], $R(s)^T \dot{R}(s) \in so(3)$ est une matrice 3x3 antisymétrique ($so(3)$ est l'algèbre de Lie du groupe de Lie $SO(3)$). Sachant que l'ensemble des matrices antisymétriques est isomorphe à \mathbb{R}^3 , on définit l'application bijective de \mathbb{R}^3 dans $so(3)$ pour $\mathbf{u} = [u_x, u_y, u_z]^T \in$

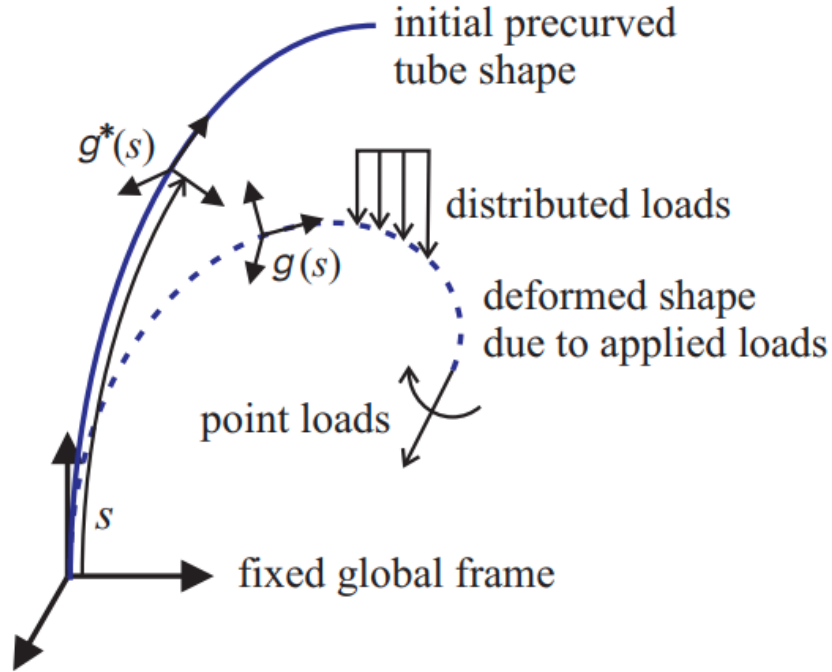


FIGURE 2.14 – Repère $g^*(s)$ lié à l'état de référence et repère $g(s)$ lié à l'état déformé sous l'action d'efforts externes - [Rucker 2010]

\mathbb{R}^3 :

$$\hat{u} = \begin{bmatrix} 0 & -u_z & u_y \\ u_z & 0 & -u_x \\ -u_y & u_x & 0 \end{bmatrix} \quad (2.4)$$

On a également la réciproque : $\hat{u}^\vee = u$.

De même, $g(s)^{-1}\dot{g}(s) \in se(3)$ (l'algèbre de Lie de $SE(3)$) est isomorphe à \mathbb{R}^6 de sorte que pour $\hat{\xi} = [v_x, v_y, v_z, u_x, u_y, u_z]^T \in \mathbb{R}^6$:

$$\hat{\xi} = \begin{bmatrix} 0 & -u_z & u_y & v_x \\ u_z & 0 & -u_x & v_y \\ -u_y & u_x & 0 & v_z \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.5)$$

avec $\hat{\xi}^\vee = \xi$.

Si l'on possède un repère paramétré, on peut ainsi obtenir un vecteur contenant les taux de variation de $g(s)$ par rapport à s exprimé dans le repère

de coordonnées de $g(s)$.

$$\xi = [v_x, v_y, v_z, u_x, u_y, u_z]^T = (g^{-1}(s)\dot{g}(s))^\vee, \quad (2.6)$$

Les vecteurs $v = [v_x, v_y, v_z]^T$ et $u = [u_x, u_y, u_z]^T$ forment ainsi respectivement les taux de variations linéaires et angulaires de $g(s)$.

De manière équivalente, si l'on connaît le vecteur $\xi(s)$ ainsi que le repère initial $g(0)$, on peut intégrer l'équation suivante pour obtenir $\dot{g}(s)$ le long de la tige :

$$\dot{g}(s) = g(s)\widehat{\xi}(s) = \begin{bmatrix} R(s) & p(s) \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} 0 & -u_z & u_y & v_x \\ u_z & 0 & -u_x & v_y \\ -u_y & u_x & 0 & v_z \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.7)$$

Ainsi on obtient

$$\dot{p}(s) = R(s)v, \quad \dot{R}(s) = R(s)\widehat{u}(s). \quad (2.8)$$

On a arbitrairement défini l'orientation de référence dans l'équation 2.3 comme $R^*(s)\mathbf{e}_3 = \frac{\dot{p}^*(s)}{\|\dot{p}^*(s)\|}$. En remplaçant R dans l'équation 2.8, on obtient $v^*(s) = [0, 0, \|\dot{p}^*(s)\|]^T$. Si le domaine de définition de la configuration de référence est borné, on peut alors appliquer un changement de variable de sorte à définir le paramètre s compris entre 0 et la longueur d'arc de la tige dans sa configuration de référence. On a donc $v^*(s) = [0, 0, 1]^T$. Dans le cas d'une tige sans pré-courbure, on a $R^*(s) = Id$ et en remplaçant dans 2.8, on obtient alors $u^*(s) = [0, 0, 0]^T$. Dans cette thèse, on fait l'hypothèse que l'ensemble des objets manipulés ont une courbure de référence nulle.

2.3.2 Equilibre statique

On présente dans cette partie les équations d'équilibre classiques pour une poutre Cosserat de longueur l telles que présentées dans [Antman 2005]. Soit un tronçon de la tige $[c, s]$, on note les forces et moments internes exercés par $[s, l]$ sur $[c, s]$, $n(s)$ et $m(s)$. De même, les forces et moments exercés par $[0, c]$ sur $[c, s]$ sont notés $-n(c)$ et $-m(c)$ (voir fig 2.15). La somme des efforts appliqués sur la section $[c, s]$ donne alors :

$$n(s) - n(c) + \int_c^s f(\sigma)d\sigma = 0, \quad (2.9)$$

$$m(s) + p(s) \times n(s) - m(c) - p(c) \times n(c) + \int_c^s (p(\sigma) \times f(\sigma) + l(\sigma)) d\sigma = 0, \quad (2.10)$$

Où f et l sont les forces et moments extérieurs distribués par unité de longueur sur la tige. En dérivant ces équations par rapport à s on obtient les formes classiques des équations différentielles de Cosserat gouvernant les forces et moments internes n et m dans une tige :

$$\dot{n}(s) + f(s) = 0, \quad (2.11)$$

$$\dot{m}(s) + \dot{p}(s) \times n(s) + l(s) = 0, \quad (2.12)$$

2.3.3 Sens physique des taux de variation

En appliquant des forces et moments extérieurs le long de la tige, ou bien en changeant les conditions aux limites, celle-ci va se déformer et passer de son état de référence $g^*(s)$ à un nouvel état $g(s)$. Cela implique un changement équivalent des taux de variation v^* et u^* qu'on définit de la manière suivante : $\Delta\xi(s) = \xi(s) - \xi^*(s) = [\Delta v(s)^T, \Delta u(s)^T]^T$.

Ce changement représente directement les contraintes mécaniques appliqués le long de la tige. Δv_x et Δv_y représentent les contraintes de cisaillement tandis que Δv_z représente les contraintes liées aux efforts de traction et de compression. De manière similaire, Δu_x et Δu_y sont les contraintes liées aux moments fléchissant et Δu_z celles liées à la torsion de la poutre. On suppose les sections transversales de la tige rigides.

2.3.4 Loi de comportement

On a jusqu'ici les équations différentielles reliant v et u à p et R et celles reliant p et R à n et m . Il nous manque donc une expression reliant v et u à n et m .

On définit les matrices de raideur linéaires et angulaires K_t et K_r :

$$K_t = \begin{bmatrix} A(s)G & 0 & 0 \\ 0 & A(s)G & 0 \\ 0 & 0 & A(s)E \end{bmatrix} \quad (2.13)$$

$$K_r = \begin{bmatrix} I_x(s)E & 0 & 0 \\ 0 & I_y(s)E & 0 \\ 0 & 0 & J(s)G \end{bmatrix}$$

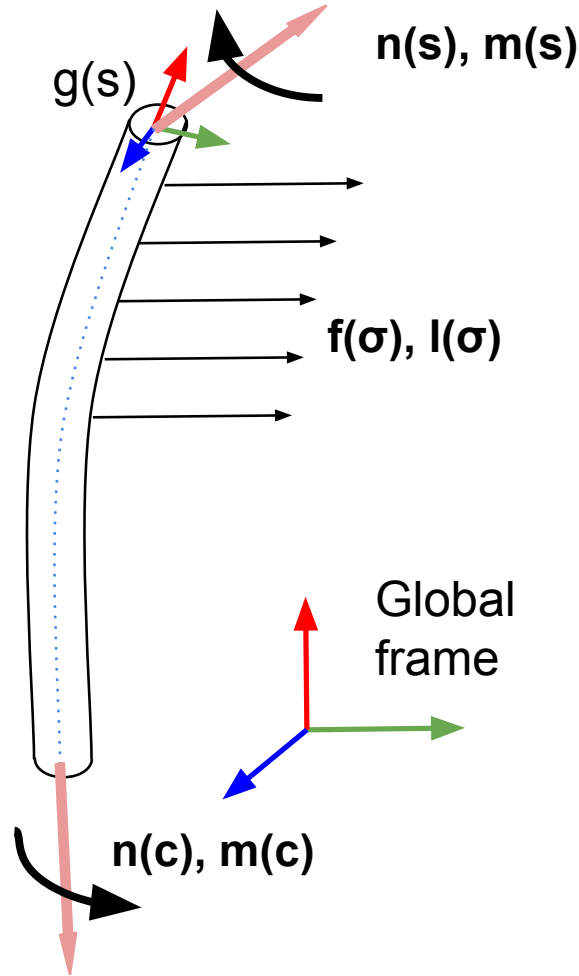


FIGURE 2.15 – Section $[c, s]$ de la tige sujette à des forces et moments extérieurs f, l , distribués le long de la section.

Où I_x et I_y sont les moments d'inertie de la section transversale de surface A , J le moment d'inertie selon l'axe z . E et G sont les modules d'Young et de cisaillement du matériau. A partir de la loi de Hooke : $F = k\Delta x$ et des matrices de raideur, on peut ainsi définir l'expression de n et m exprimés dans le repère des coordonnées locales :

$$\begin{aligned} n(s) &= R(s)K_t(s)(v(s) - v^*(s)), \\ m(s) &= R(s)K_r(s)(u(s) - u^*(s)) \end{aligned} \quad (2.14)$$

De 2.14, on déduit directement l'expression de v et u en fonction de s :

$$\begin{aligned} v(s) &= K_t^{-1} R^T(s) n(s) + v^*(s), \\ u(s) &= K_r^{-1} R^T(s) m(s) + u^*(s) \end{aligned} \quad (2.15)$$

2.3.5 Equations de comportement

A partir des équations 2.11 et 2.12, et en dérivant les équations 2.15 par rapport à s , on obtient ainsi le système complet d'équations différentielles permettant de calculer la forme de la tige :

$$\begin{aligned} \dot{p} &= Rv, \\ \dot{R} &= R\hat{u}, \\ \dot{v} &= \dot{\tilde{v}} - K_t^{-1}(\hat{u}K_t\Delta v + R^T f), \\ \dot{u} &= \dot{\tilde{u}} - K_r^{-1}(\hat{u}K_r\Delta u + \hat{v}K_t\Delta v + R^T l) \end{aligned} \quad (2.16)$$

En utilisant m et n comme variables d'état plutôt que v et u , on obtient le système équivalent :

$$\begin{aligned} \dot{p} &= Rv, & v &= K_t^{-1} R^T n + \tilde{v}, \\ \dot{R} &= R\hat{u}, & u &= K_r^{-1} R^T m + \tilde{u}, \\ \dot{n} &= -f, \\ \dot{m} &= -\dot{p} \times n - l \end{aligned} \quad (2.17)$$

On utilise le système 2.17 dans cette thèse. Si on connaît p , R , n et m en $s = 0$, on peut ainsi intégrer directement ces équations pour obtenir p , R , n et m par rapport à s le long de la tige.

Génération de trajectoire pour la manipulation d'un objet déformable linéaire

3.1 Introduction et objectifs

3.1.1 Cadre de contrôle de robots déformables

Dans le chapitre précédent, nous avons étudié les différentes approches robotiques utilisant le modèle Cosserat. L'écrasante majorité de celles-ci ont été développées pour modéliser des robots déformables, réels ou simulés.

Lors du contrôle de robots déformables linéaires [Rucker 2010, Rucker 2011, Campisano 2021, Boyer 2022], le principal objectif est de contrôler la position, l'orientation ou la trajectoire de l'extrémité du robot. Dans la littérature associée, le caractère déformable du robot est principalement utilisé pour permettre à l'extrémité d'atteindre des configurations qui seraient inaccessibles à un robot rigide. Dans ces travaux, les auteurs déterminent la solution des équations constitutives du modèle Cosserat (voir équations annexe 2.15) qui respectent les consignes imposées sur la position et l'orientation de l'extrémité du robot. Même dans les travaux où l'entièreté du corps déformable du robot est utilisée pour la manipulation [Coevoet 2019], la forme du robot n'est pas définie directement. Sur l'exemple en figure 3.1, les auteurs contrôlent un ensemble de points sur l'objet rigide à l'aide d'un manipulateur linéaire déformable.

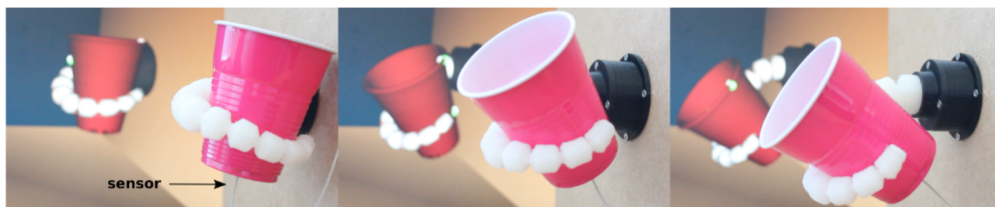


FIGURE 3.1 – Exemple de manipulation d'objet avec un robot déformable [Coevoet 2019]

Bien que ces approches soient efficaces et adaptées aux défis soulevés dans la communauté de contrôle de robots déformables, elles sont peu adaptées au contrôle précis de la forme d'objets. La manipulation d'objets déformables soulève un problème différent mais avec certaines similarités. Dans les deux cas, il est nécessaire de résoudre un problème sous actionné, le nombre d'actionneurs étant nécessairement fini tandis que le nombre de degrés de liberté de l'objet ou du robot ne l'est pas. Toutefois pour le contrôle précis de la forme d'un objet déformable, il est alors nécessaire d'ajouter davantage de contraintes sur la forme de l'objet, tout en s'assurant que ces contraintes n'entrent pas en opposition les unes avec les autres. En contrepartie, le problème de modélisation et conception des actionneurs est beaucoup plus simple puisqu'il n'est pas nécessaire de concevoir la stratégie de contrôle du robot en fonction du modèle de l'objet.

3.1.2 Objectif

Dans ce chapitre, nous présentons notre méthode pour générer une trajectoire du robot permettant de résoudre une tâche de manipulation en prenant en compte les contraintes appliquées sur l'objet. Dans un premier temps, nous adaptons le cadre présenté dans le chapitre précédent et développé dans [Rucker 2010, Rucker 2011, Rucker 2013] pour le contrôle de robot déformable à la manipulation d'objets linéaires. Pour cela, nous allons reformuler le problème de sorte à ne plus contrôler la position de l'extrémité d'un robot, mais plutôt la forme complète de l'objet. De plus, dans le cas des robots déformables, le problème aux limites est généralement formulé comme un problème de poutre encastré-contrôlé (voir figure 3.2). On considère la condition aux limites contrôlée comme un type d'encastrement où la position et l'orientation du repère associé au point d'encastrement est contrôlée par le robot au cours du temps. Nous reformulons donc le problème classique pour prendre en compte un ensemble plus vaste de conditions aux limites, typiquement des conditions contrôlé-contrôlé, mais également des conditions où certains degrés de liberté sont libres aux limites tandis que d'autres sont contrôlés.

Cette reformulation implique que la solution au problème inverse n'est plus unique. On définit donc un nouvel objectif à partir de la théorie de la commande optimale [Boyer 2022]. Ce deuxième objectif permet ainsi de minimiser l'énergie élastique dans l'objet manipulé. Contrairement aux auteurs dans [Boyer 2022], on ne cherche pas à résoudre le problème direct en trouvant la forme stable de l'objet, soumis aux conditions aux limites,

Préhenseur robotique

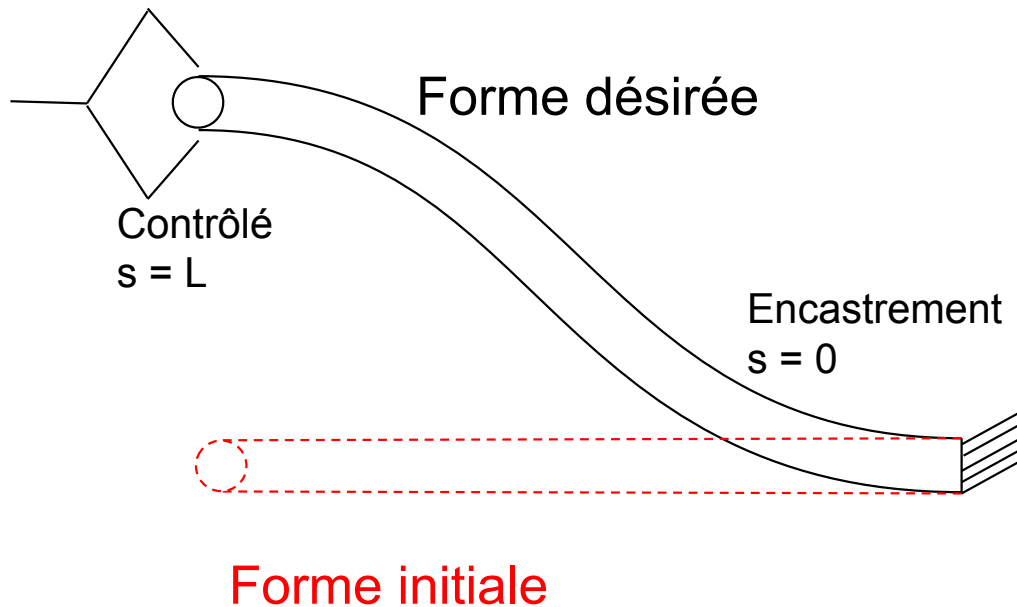


FIGURE 3.2 – Conditions aux limites d’une tige contrôlée-encastree pour une tâche de contrôle de forme

qui minimise l’énergie élastique. Dans notre cas, il existe un ensemble de solutions qui satisfassent l’objectif en position. On cherche ainsi, parmi ces configurations, celle qui minimise également l’énergie élastique dans l’objet. Cette énergie élastique est directement liée aux efforts du préhenseur appliqués à l’objet. En résolvant le BVP avec la méthode de tir, on trouve ainsi une configuration de l’objet qui minimise à la fois l’erreur en position et l’énergie élastique dans l’objet, et donc le risque d’endommager l’objet.

On définit ensuite un ensemble d’objectifs intermédiaires. Chaque problème sera résolu itérativement de sorte à éviter les minima locaux et à obtenir une trajectoire de l’objet manipulé. On définit ainsi une trajectoire robuste et répétable permettant d’atteindre un objectif de forme de manière précise tout en minimisant les risques d’endommager l’objet manipulé. On testera ces trajectoires pour une tâche de contrôle de forme d’un objet déformable linéaire avec un bras robotique.

3.2 Formulation du problème à objectifs multiples

3.2.1 Forme désirée

On considère la configuration totale de la bande Γ comme l'ensemble des configurations de chaque rigide infinitésimal constituant la tige linéaire, soit la position p , la matrice de rotation R , les efforts et moments internes n et m par rapport à s . Γ est obtenu en intégrant spatialement les équations différentielles du modèle Cosserat définies dans l'équation 3.1 et l'annexe 2.3, à partir de la valeur initiale Γ_0 , où $\Gamma_0 = [p(0), R(0), n(0), m(0)]^T$. Cela signifie donc que le type de conditions aux limites ainsi que la valeur initiale Γ_0 en $s = 0$ permet de déterminer entièrement Γ pour tout s entre 0 et la longueur de la tige L .

Dans ce chapitre, nous allons appliquer le cadre mathématique présenté dans l'annexe 2.3. On rappelle les équations constitutives du modèle Cosserat :

$$\begin{aligned}
 \dot{p} &= Rv, & v &= K_t^{-1} R^T n + \tilde{v}, \\
 \dot{R} &= R\hat{u}, & u &= K_r^{-1} R^T m + \tilde{u}, \\
 \dot{n} &= -f, \\
 \dot{m} &= -\dot{p} \times n - l
 \end{aligned} \tag{3.1}$$

Ces équations 3.1 régissent l'état de l'objet $\Gamma = [p, R, n, m]^T$ par rapport à s . Soit la position p , la matrice de rotation R , les forces et moments internes n et m le long de l'objet.

On définit une représentation bas niveau de la forme souhaitée de l'objet qui consiste en un ensemble de n points objectifs p_{obj}^i , $i \in [1, n_c]$. Cette représentation classique permet de fortement réduire la dimension du problème en réduisant la forme désirée à certains points objectifs clés judicieusement choisis. Le nombre et la position de ces points objectifs dépend entièrement de la tâche. En règle générale, plus la forme souhaitée est complexe, plus les modes de déformation de l'objet sont excités, et plus dense sera l'ensemble des points objectifs. Cette densité peut être locale ou globale sur la forme entière de l'objet. Cette formulation est très répandue, que ce soit pour le contrôle de robots déformables [Rucker 2010], la manipulation d'objets déformables [Ficuciello 2018, Duriez 2016, Shetab-Bushehri 2022, Koessler 2021] ou dans un tout autre domaine, l'asservissement visuel [Chaumette 2006, Navarro-Alarcon 2014]. Toutefois dans la formulation classique, à chaque point objectif p_{obj}^i est associé un point de contrôle p_c^i attaché au repère de l'objet, et l'erreur en position ou orientation entre ce point objectif et ce point de contrôle est minimisée. Le but étant de faire correspondre la forme de l'objet à la forme définie par p_{obj}^i .

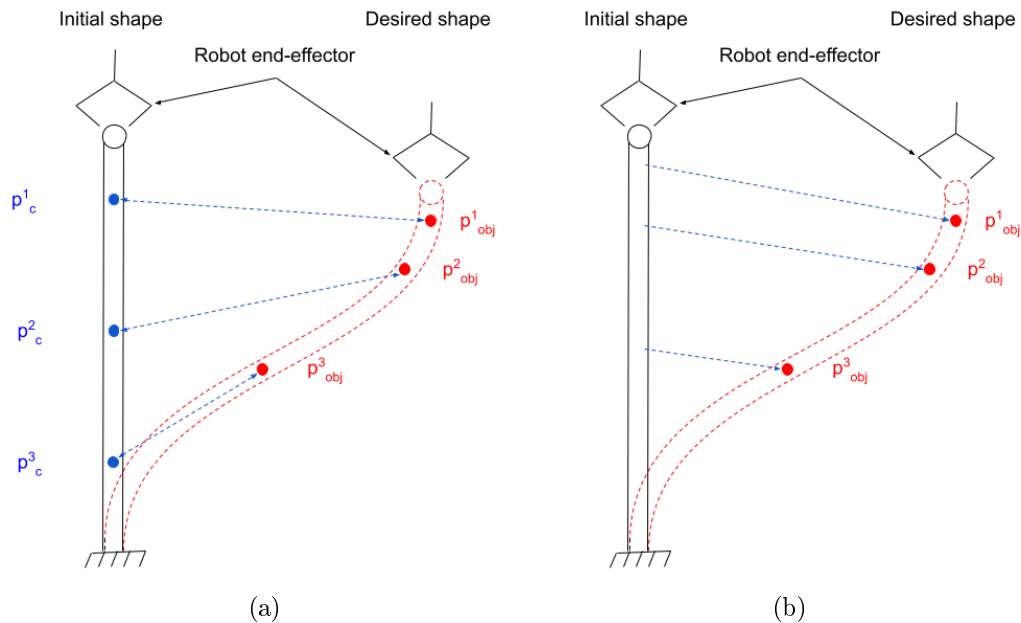


FIGURE 3.3 – Cas d’échec pour une forme atteignable avec l’approche classique point objectif-contrôle

Cette approche est particulièrement bien adaptée lorsque le nombre de points de contrôle est inférieur ou égal au nombre de points actionnés, par exemple dans le cas du contrôle de l’extrémité d’un robot linéaire. Toutefois, pour la manipulation d’objets linéaires, les seuls points actionnés sont ceux saisis par le robot. Dans la majeure partie des cas cela signifie au plus 1 ou 2 points de contrôle, ce qui est loin d’être assez pour atteindre des formes complexes.

Une illustration d’un cas d’échec avec la formulation classique est proposée sur la figure 3.3a. On a une tige encastrée à une extrémité et saisie par le robot à l’autre extrémité. Pour faire correspondre p_c^i à p_{obj}^i , il faudrait appliquer simultanément, à l’aide d’un seul point de saisie, une compression sur la partie supérieure de la tige, et une extension sur la partie inférieure, ce qui est impossible. Un algorithme basé sur cette approche ne trouverait pas de solution quand bien même la forme désirée est tout à fait atteignable dans les conditions présentées (voir figure 3.3b). Cette situation est très fréquente et se produit lorsque les contraintes de couplage entre différents points de l’objet ne sont pas respectées, et ces contraintes sont compliquées à exprimer analytiquement, d’autant plus lorsque le nombre de points objectifs est élevé.

Nous avons réfléchi à deux manières de contourner ce problème :

- La première et la plus évidente consiste à s’assurer expérimentalement

que les contraintes de couplage sont respectées. La manière la plus simple est d'amener à la main l'objet dans la configuration souhaitée, mesurer la position des points de contrôle, puis assigner ces valeurs aux points objectifs. On s'assure de fait que les contraintes seront respectées et qu'une trajectoire de l'objet qui amène chaque point de contrôle vers son point objectif existe. Pour une application où l'on tente d'atteindre une forme spécifique, cette manière est envisageable. Mais dans un cas où l'on cherche une solution optimale selon plusieurs critères parmi un ensemble de solutions, elle n'est pas applicable.

- La seconde, adoptée dans ce chapitre, consiste à reformuler et à modifier le problème de sorte à faire disparaître ce couplage. Pour cela, on minimise la distance dans \mathbb{R}^3 entre la courbe Λ formée par chaque point de l'objet, et l'ensemble des points objectifs p_{obj}^i . Au lieu d'amener des points spécifiques de l'objet vers une position objectif, on crée ainsi une courbe qui passe par chaque point objectif. Dans les faits, cela revient à relâcher les contraintes sur les points de contrôle en permettant à n'importe quel point de la courbe de prendre ce rôle. En relâchant les contraintes, on augmente toutefois le nombre de solutions ainsi que le coût en calculs puisque la distance entre chaque point de la courbe et chaque point objectif doit être calculée.

La distance entre la courbe $\Lambda = [\alpha(s), \beta(s), \gamma(s)]^T$ et $p_{obj}^i = [p_{obj_x}^i, p_{obj_y}^i, p_{obj_z}^i]^T$ est exprimée dans l'équation 3.2 :

$$\begin{aligned}
 D &= \sum_{i=1}^n dist(\Lambda, p_{obj}^i) \\
 &= \sum_{i=1}^n \sqrt{\min_s ((\alpha(s) - p_{obj_x}^i)^2 + (\beta(s) - p_{obj_y}^i)^2 + (\gamma(s) - p_{obj_z}^i)^2)}
 \end{aligned} \tag{3.2}$$

3.2.2 Énergie de déformation

Afin de préserver l'objet manipulé, on introduit l'énergie de déformation au sein de l'objet. Des concepts similaires ont déjà été utilisés précédemment, dans [Duenser 2018] les auteurs proposent de calculer l'énergie d'un maillage d'éléments finis afin de trouver une position d'équilibre stable. Dans [Boyer 2022], les auteurs résolvent le problème cinématique direct en minimisant l'énergie élastique dans l'objet, soumis à des conditions aux limites.

Dans notre cas, la solution n'est pas unique. En effet, en augmentant le

nombre d'inconnues si l'on considère le cas d'un objet manipulé à ses deux extrémités, le problème de Cosserat inverse n'est plus bien posé. Dans la majorité des cas de contrôle de robots déformables, le robot est encastré à sa base. Dans ce cas, le vecteur des conditions initiales inconnues est $\Gamma_0 = [n(0), m(0)]^T$ car la position et l'orientation sont fixées en $s = 0$, le nombre d'inconnues est ainsi un vecteur de \mathbb{R}^6 pour une application en trois dimensions, et de \mathbb{R}^3 en deux dimensions. Le nombre d'équations est quant à lui déterminé par les contraintes sur la position et l'orientation en $s = L$. Il y a alors, en trois dimensions, trois équations sur la position et trois sur l'orientation.

Toutefois si l'objet est saisi à ses deux extrémités par deux bras robotiques, la position et la matrice de rotation en $s = 0$ deviennent également des inconnues. $\Gamma_0 = [p(0), R(0), n(0), m(0)]^T$ est ainsi un vecteur de \mathbb{R}^{12} en trois dimensions. De plus, comme présenté dans la section précédente, on ne contraint plus l'orientation des points objectifs. Il existe donc un ensemble de solutions qui satisfont l'objectif en position. On cherche ainsi, parmi ces solutions, celle qui minimise également l'énergie élastique dans l'objet.

Sans efforts extérieurs autres que ceux inclus dans le modèle, cette énergie dépend uniquement des efforts et moments internes dans l'objet. En supposant le système quasi-statique, on définit l'énergie élastique dans l'objet de la manière suivante (voir equation 3.3) :

$$\begin{aligned} E &= \int_0^L e(s) ds, \\ &= \int_0^L \left(\frac{1}{2} (v(s) - \tilde{v}(s))^T K_{tra}(s) (v(s) - \tilde{v}(s)) \right. \\ &\quad \left. + \frac{1}{2} (u(s) - \tilde{u}(s))^T K_{rot}(s) (u(s) - \tilde{u}(s)) \right) ds \end{aligned} \quad (3.3)$$

En l'absence de contacts extérieurs non modélisés, l'énergie de déformation dépend ainsi uniquement des taux de variation linéaire et angulaire v et u , qui sont directement liés aux efforts et moments internes (voir equation 3.1).

On peut toutefois noter que bien que le problème Cosserat inverse ne soit plus bien posé, le problème direct est toujours bien posé peu importe la formulation. Dans le problème direct, Γ_0 est toujours connu que l'objet soit encastré en $s = 0$ ou manipulé par un robot à ses deux extrémités, et les inconnues sont $[p, R, n, m]^T$ en tout autre point de l'objet. Il y a donc toujours douze inconnues et 12 équations constitutives (équation 3.1) qui gouvernent le comportement de l'objet.

3.2.3 Fonction de coût

Intégrer les équations de Cosserat à partir de Γ_0 nous donne la configuration totale de l'objet. Le problème peut donc être réduit à : Trouver les valeurs initiales Γ_0 qui, une fois intégrées, fournissent une configuration Γ qui minimise la fonction de coût.

La forme des variables d'optimisation Γ_0 dépend du type de conditions aux limites. Pour une poutre encastree-contrôlée, $\Gamma_0 = [n(0), m(0)]^T$, et pour une poutre contrôlée-contrôlée, on rajoute la position et l'orientation à la base de l'objet dans les variables et $\Gamma_0 = [p(0), R(0), n(0), m(0)]^T$.

La fonction de coût doit refléter les deux objectifs : forme et énergie, mais pas nécessairement avec des poids équivalents. On cherche à définir les contraintes minimales sur la forme de sorte à s'assurer l'ensemble le plus vaste possible de solutions permettant d'accomplir la tâche souhaitée. On recherche alors la solution, ou une solution proche, qui offre le meilleur compromis entre minimisation de l'énergie et minimisation de l'objectif de forme.

Pour un objet linéaire de longueur L , à partir des équations 3.2 et 3.3, on définit la fonction de coût suivante :

$$\begin{aligned}
 \min_{n(0), m(0)} F &= W_1 E + W_2 D, \\
 &= \frac{W_1}{2} \int_0^L (v(s) - \tilde{v}(s))^T K_{tra}(s) (v(s) - \tilde{v}(s)) + (u(s) - \tilde{u}(s))^T \\
 &K_{rot}(s) (u(s) - \tilde{u}(s)) ds + \\
 &W_2 \sum_{i=1}^n \sqrt{\min_s ((\alpha(s) - p_{obj_x}^i)^2 + (\beta(s) - p_{obj_y}^i)^2 + (\gamma(s) - p_{obj_z}^i)^2)},
 \end{aligned} \tag{3.4}$$

sous :

$$\left\{ \begin{array}{l} p(s=L) \subset W_s, \\ \dot{p} = Rv, \\ \dot{R} = R\hat{u}, \\ \dot{n} = -f, \\ \dot{m} = -\dot{p} \times n - l \end{array} \right.$$

W_1 et W_2 sont les poids appliqués à chaque terme, et W_s représente l'espace de travail opérationnel du robot (voir figure 3.4).

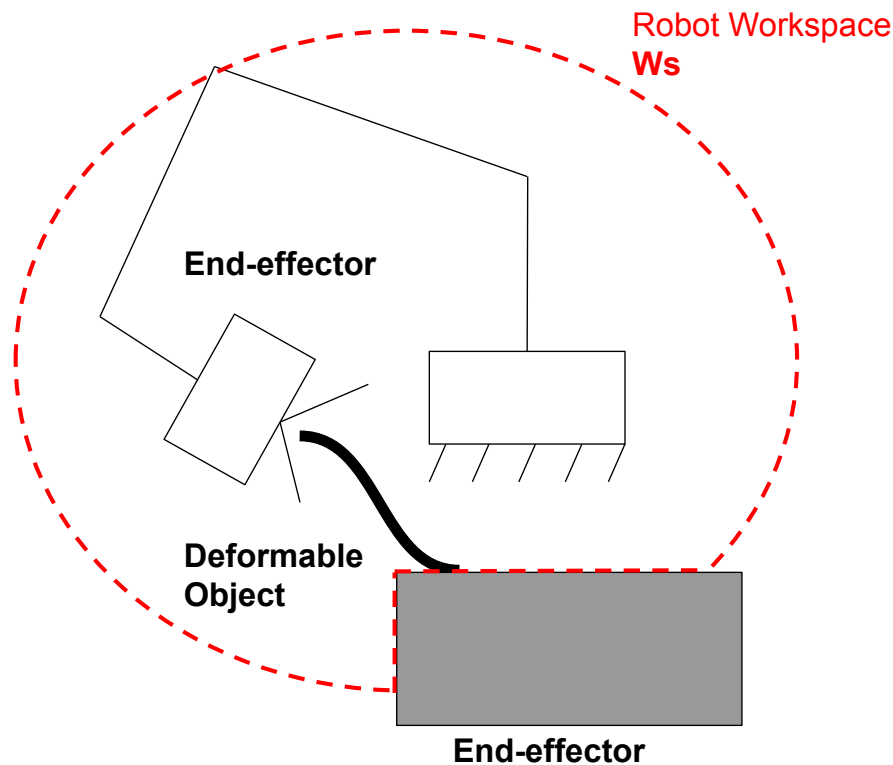


FIGURE 3.4 – Représentation schématique du dispositif expérimental

3.3 Résolution numérique et génération de trajectoire

3.3.1 Méthode de résolution

Résoudre ce problème d'optimisation implique de résoudre le problème cinématique inverse. Le problème n'est pas trivial, une solution analytique au modèle Cosserat inverse est difficile à trouver. On doit donc résoudre le problème numériquement. On utilise un algorithme inspiré de la méthode de tir [Rao 2009]. Cette méthode permet de résoudre un problème aux valeurs limites en le réduisant à un problème aux valeurs initiales. Différentes valeurs initiales vont être testées jusqu'à trouver une solution qui satisfasse également les conditions aux limites. Dans notre cas, les conditions limites sont représentées par la fonction de coût, et les valeurs initiales Γ_0 sont mises à jour par l'algorithme jusqu'à trouver une solution intégrée qui minimise la fonction de coût. Un schéma représentatif du processus est présenté sur la figure 3.5.

La manière dont sont corrigées les conditions initiales dépend du type

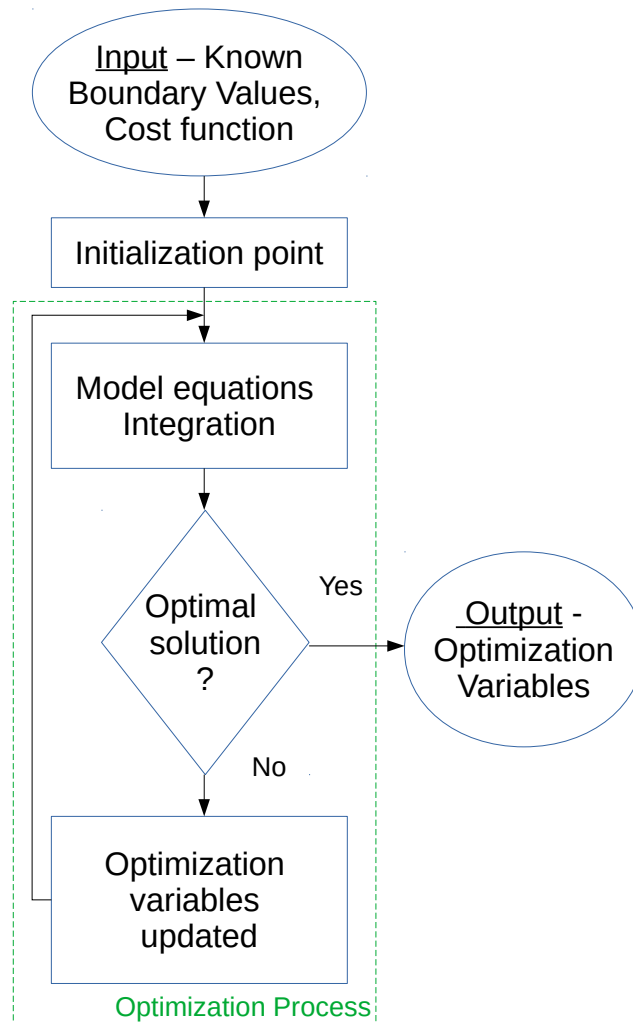
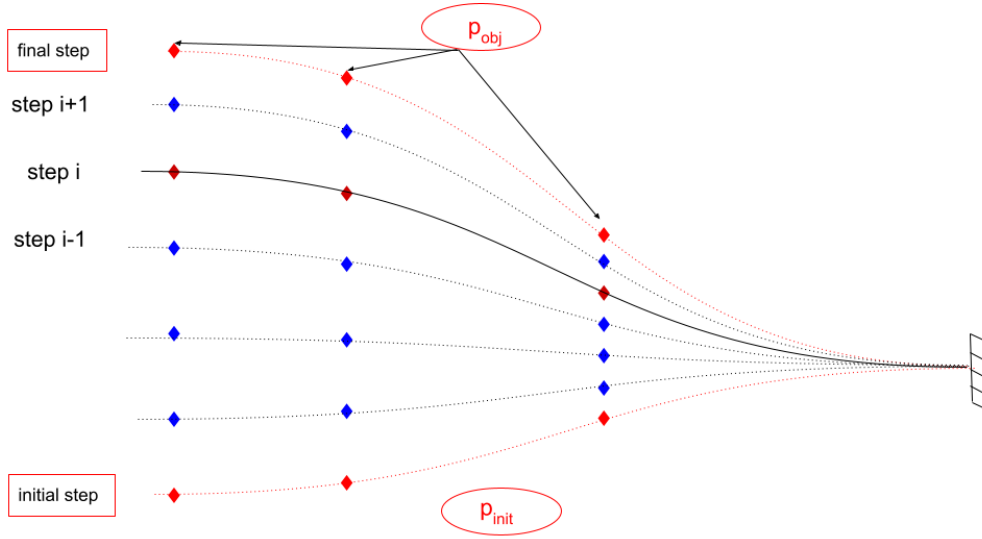


FIGURE 3.5 – Processus d’optimisation

d’algorithme d’optimisation choisi. Dans ce chapitre, nous utilisons un algorithme basé sur la méthode d’optimisation quadratique successive (SQP [Nocedal 2006]) puisque le nombre de variables d’optimisation est relativement faible. La méthode du point intérieur (interior point method [Potra 2000]) et l’algorithme de Levenberg-Marquardt (LM) [Moré 2006] donnent des performances similaires.

3.3.2 Minima locaux

SQP LM et Interior point sont des méthodes de recherche locales, elles sont donc susceptibles de trouver des solutions locales. Or, avec les objets

FIGURE 3.6 – Ensemble des n_{inter} configurations successives

déformables ou tout autre problème avec de nombreuses non-linéarités, ces solutions locales sont très nombreuses.

Fournir à l'algorithme un point d'initialisation proche de la solution permet d'accélérer la recherche et de réduire la probabilité de tomber dans un minimum local. Cela présuppose toutefois que l'on connaît déjà, ou au moins que l'on a une bonne idée de ce à quoi ressemble la solution, ce qui est rarement le cas.

Afin d'éviter les minima locaux et de trouver une solution globale, on propose de résoudre itérativement le problème en partant de la configuration initiale de la bande. Soit p_{init}^i la projection de p_{obj}^i sur la configuration initiale de la bande. En discrétisant l'objectif en forme entre p_{init}^i et p_{obj}^i , on introduit un ensemble de n_{inter} objectifs intermédiaires (voir équation 3.5). p_{init}^i étant connu, on résout le problème d'optimisation de proche en proche à chaque étape intermédiaire en utilisant la solution du problème précédent comme point d'initialisation de l'étape suivante. Plus les écarts entre chaque configuration intermédiaire est faible, plus basses sont les chances de tomber dans des minima locaux.

Les objectifs intermédiaires sont définis dans l'équation suivante :

$$p_{inter_j}^i = p_{init}^i + (j - 1) \frac{p_{obj}^i - p_{init}^i}{n_{inter}}, \quad (3.5)$$

Où $i \in [1, n_c]$, n_c est le nombre de points formant la configuration objectif, et $j \in [1, n_{inter}]$, n_{inter} est le nombre d'étapes intermédiaires.

3.3.3 Génération de trajectoires

La méthode de résolution présentée précédemment a un autre avantage, elle nous permet de générer la trajectoire du robot. Les solutions intermédiaires sont, de fait, des configurations intermédiaires de la bande entre les configurations initiales et finales.

Dans [Campisano 2021], les auteurs définissent la trajectoire en discrétisant la position et l'orientation du manipulateur entre la configuration initiale et la configuration finale calculée. La trajectoire est donc une ligne droite entre deux positions. Avec cette approche, non seulement l'état de l'objet n'est pas connu au cours de la manipulation, mais on n'a pas de certitude que le chemin emprunté est faisable.

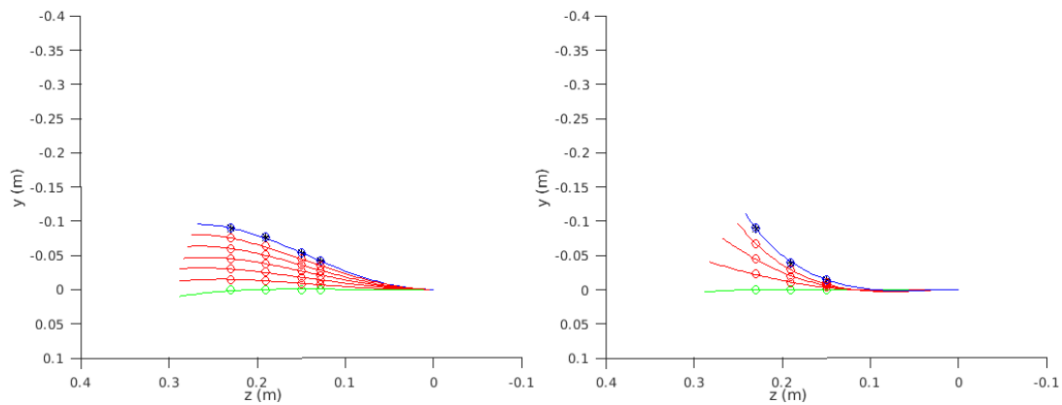
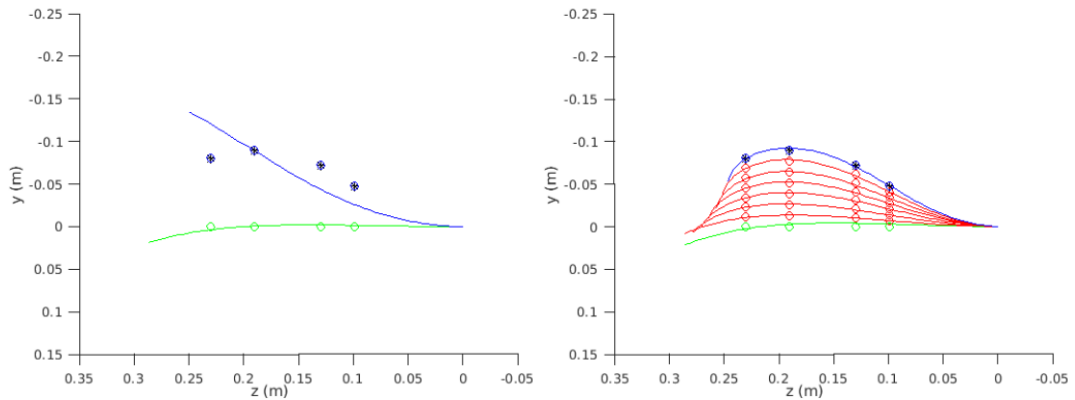


FIGURE 3.7 – Exemple de trajectoires générées :

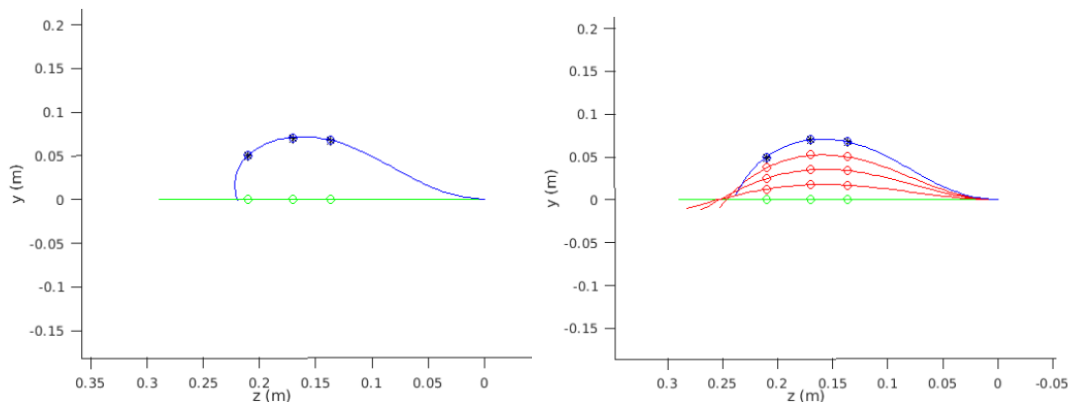
la configuration finale de l'objet est représentée en bleu, les cercles bleus correspondent à p_{obj}^i . La configuration initiale est représentée en vert, les cercles verts représentent p_{init}^i . Les configurations intermédiaires sont représentées en rouge, les cercles rouges représentent $p_{inter_j}^i$

En utilisant les configurations intermédiaires comme points de passage, on s'assure non seulement que le chemin est faisable, mais également que la fonction de coût est minimisée à chaque étape, les efforts appliqués sur l'objet sont donc minimisés pendant toute la tâche. D'autant plus que pour une application réelle, certaines hypothèses ne sont pas vérifiées. On a par exemple supposé l'effet de la viscosité négligeable, or c'est rarement le cas en réalité. De fait, la configuration de l'objet dépend nécessairement du chemin emprunté, il y a donc un intérêt supplémentaire à contrôler la forme de l'objet du début à la fin de la tâche. Cette approche est représentée graphiquement sur la figure 3.6.

Pour générer la trajectoire du robot à partir de la trajectoire de l'objet, on récupère, à chaque étape, le vecteur des configurations Γ des points de



(a) Configuration finale de l'objet lorsque l'algorithme tombe dans un minimum local
 (b) Solution trouvée avec une résolution itérative



(c) Trajectoire infaisable avec l'approche classique - Configuration chaotique
 (d) Trajectoire proposée - évite les configurations d'équilibre instable

FIGURE 3.8

l'objet, aux points de saisie. La consigne à envoyer au robot dépend du type de contrôle. Pour une commande en force, on utilise les forces et moments internes n et m , tandis que pour une commande en position, on récupère la position et la matrice de rotation p et R . On passe du chemin à la trajectoire en appliquant une vitesse constante au robot, ce qui nous donne un temps à chaque étape. On suppose le problème quasi-statique, la vitesse du robot doit donc être assez basse pour respecter cette hypothèse.

On montre sur la figure 3.6 des exemples de trajectoires générées pour un objet linéaire encastré à droite, et manipulé par le robot à sa limite gauche. On y représente en bleu la forme finale de l'objet, où les cercles bleus correspondent aux points objectifs. La courbe verte correspond à la forme de l'objet à l'état initial et les cercles verts sont la projection des points objectifs sur la forme initiale. Les courbes rouges correspondent aux

formes intermédiaires de l'objet, et les cercles rouges représentant les objectifs intermédiaires sont obtenus en discrétisant spatialement entre p_{obj}^i et p_{init}^i .

Les problèmes précédemment mis en avant sont également montrés sur la figure 3.8. Sur les figures 3.8a et 3.8b, en cherchant directement la configuration finale de l'objet, l'algorithme tombe dans un minimum local et ne trouve pas de configuration qui minimise de manière satisfaisante l'erreur de forme. On voit que la forme proposée ne passe que par un seul des points objectifs au lieu des quatre. Sur les figures 3.8c et 3.8d, l'algorithme trouve bien une solution acceptable, mais la trajectoire obtenue avec une approche classique, c'est-à-dire en discrétisant la position du manipulateur entre les positions initiales et finales, n'est pas faisable. Cette trajectoire implique de déplacer le manipulateur le long de l'axe de l'objet en le comprimant, le mettant dans une position d'équilibre instable [O'Reilly 2011] (voir figure 3.9). Sur la figure 3.8d, on observe que la trajectoire de l'objet que l'on propose évite les configurations de flambement.

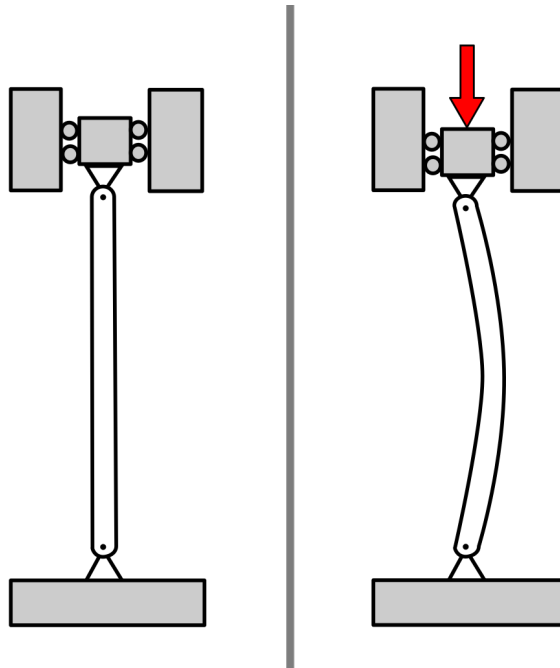


FIGURE 3.9 – Exemple de flambement, configuration chaotique où l'objet peut fléchir d'un côté ou de l'autre. Illustration tirée de la page wikipedia sur le flambement

Le nombre n_{inter} de points intermédiaires joue également un rôle primordial, moins il est élevé, plus la variation entre les valeurs initiales de deux états successifs est important et donc plus l'algorithme est susceptible de tomber dans un minimum local.

On peut résumer ce processus à (voir figure 3.10) :

- Définir p_{obj}^i ;
- Définir les points objectifs intermédiaires $p_{inter_j}^i$ entre p_{init}^i connu et p_{obj}^i ;
- Résoudre le problème d'optimisation à chaque étape en utilisant la solution de l'étape précédente comme point d'initialisation ;
- Déterminer les commandes du robot à partir des configurations intermédiaires.

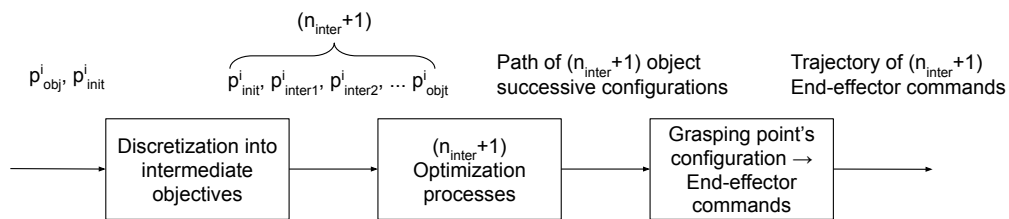


FIGURE 3.10 – Étapes successives pour obtenir une trajectoire du manipulateur à partir d'un état initial et d'une configuration objectif.

3.4 Validation expérimentale

3.4.1 Dispositif Expérimental

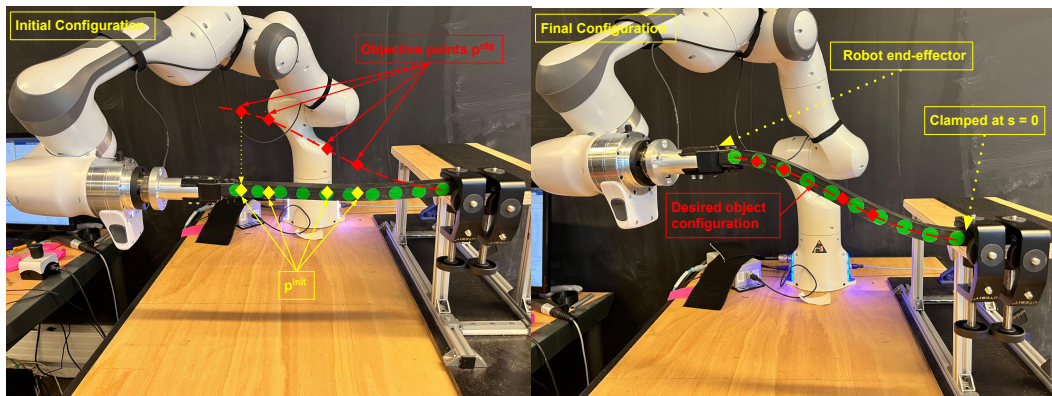


FIGURE 3.11 – Dispositif expérimental : configurations initiales et finales de l'objet.

Pour valider expérimentalement l'approche décrite précédemment, on propose la tâche suivante : déformer, à l'aide d'un bras robotique, un

objet linéaire vers une configuration souhaitée en trois dimensions tout en minimisant l'énergie de déformation dans l'objet. L'objet est encasté à une extrémité, et saisi par le bras à l'autre. Il a une section carrée qu'on suppose constante et est constitué en totalité de caoutchouc, les paramètres géométriques et élastiques de l'objet sont fournis dans le tableau 3.1. La trajectoire est définie en générant $n_{inter} = 7$ configurations intermédiaires.

Paramètre	Objet Linéaire	Bande élastique 2D
E	$3.2e^6 \text{ N/m}^2$	$3.2e^6 \text{ N/m}^2$
ν	0.5	0.5
$G = \frac{E}{2(1+\nu)}$	$1.06e^6 \text{ N/m}^2$	$1.06e^6 \text{ N/m}^2$
EI_x	0.04 N.m^2	0.8833 N.m^2
EI_y	0.04 N.m^2	0.0088 N.m^2
GJ	0.028 N.m^2	0.2974 N.m^2
ρ	1392 kg/m^3	1392 kg/m^3
L	0.29 m	0.326 m
h	0.02 m	0.01 m
b	0.02 m	0.1 m
g	9.81 m/s^2	9.81 m/s^2

TABLE 3.1 – Paramètres des objets manipulés.

On utilise un robot Franka Emika Panda avec sept degrés de liberté (voir figure 3.11). Le robot est contrôlé en position avec MOVEit et ROS. Afin d'avoir une trajectoire continue et non saccadée, on lisse spatialement le chemin du manipulateur obtenu en sortie de l'algorithme avec des B-splines. On applique des poids élevés sur les étapes initiales et finales de sorte à ne pas modifier la tâche, on a ainsi une trajectoire du robot dérivable en tout point.

Afin de respecter au mieux les hypothèses d'encastement, la bande est fixée au robot avec une poignée imprimée en 3D et une pièce en aluminium spécialement usinée pour maintenir un contact stable entre le manipulateur et la bande. Il y a toutefois de légères erreurs de conception et la position et l'orientation du manipulateur ne correspondent pas parfaitement à ceux de l'extrémité de la bande. Le setup inclut également une Kinect V2 qui suit la position de dix marqueurs répartis le long de la bande. La vision est utilisée pour mesurer la configuration de la bande au cours de la manipulation mais ne rentre pas dans le calcul de la commande.

3.4.2 Estimation des constantes élastiques

Les performances du modèle Cosserat, à l'instar des modèles physiquement réalistes, sont dépendantes de l'estimation des constantes élastiques de l'objet. Ces constantes sont le module d'Young E , qui lie la contrainte de traction et la déformation du matériau, et le coefficient de Poisson ν , qui lie la déformation transversale à la déformation longitudinale (direction de la charge, voir figure 3.12).

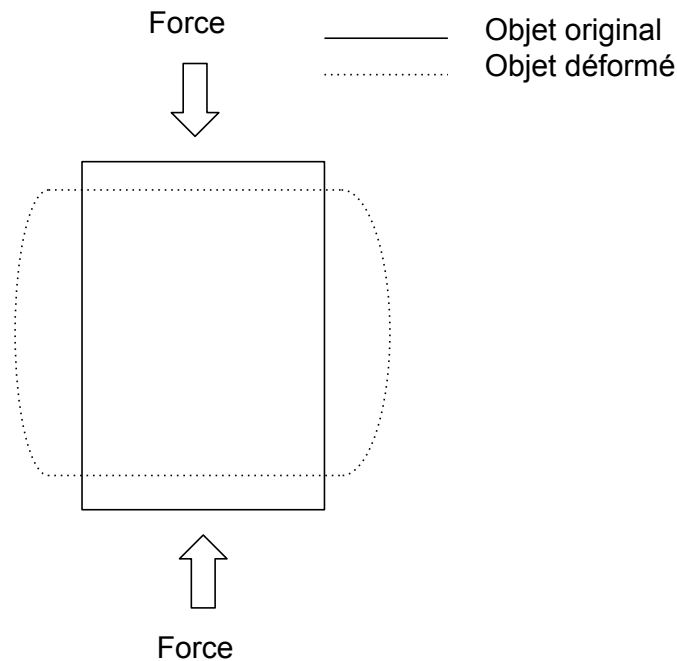


FIGURE 3.12 – Extension d'un objet dans les directions perpendiculaires à la charge, effet de Poisson.

Si le matériau est connu à l'avance, les paramètres peuvent être rentrés dans le modèle a priori. Ce n'est néanmoins pas toujours le cas et il est parfois nécessaire de les estimer. En pratique, ces paramètres sont souvent ajustés manuellement en les modifiant petit à petit jusqu'à ce que les résultats de simulation semblent visuellement plausibles par rapport à l'objet réel.

Afin d'avoir davantage de précision et en l'absence de banc d'essai en traction compression, on propose d'utiliser une approche basée sur l'optimisation des paramètres par rapport à une forme d'objet souhaitée. On encastre l'objet linéaire à une extrémité, et on laisse l'autre extrémité libre d'efforts de sorte à le laisser se déformer sous son propre poids. On récupère

ensuite à l'aide de la Kinect la position d'un ensemble de points sur l'objet qui vont former nos positions objectifs p_{obj}^i . On optimise ensuite l'erreur en position par rapport à p_{obj}^i avec comme uniques variables d'optimisation E et ν car $n(0)$ et $m(0)$ dépendent uniquement des paramètres de l'objet pour une bande encastrée - libre :

$$\min_{E, \nu} F = \sum_{i=1}^n dist(\Lambda, p_{obj}^i), \quad (3.6)$$

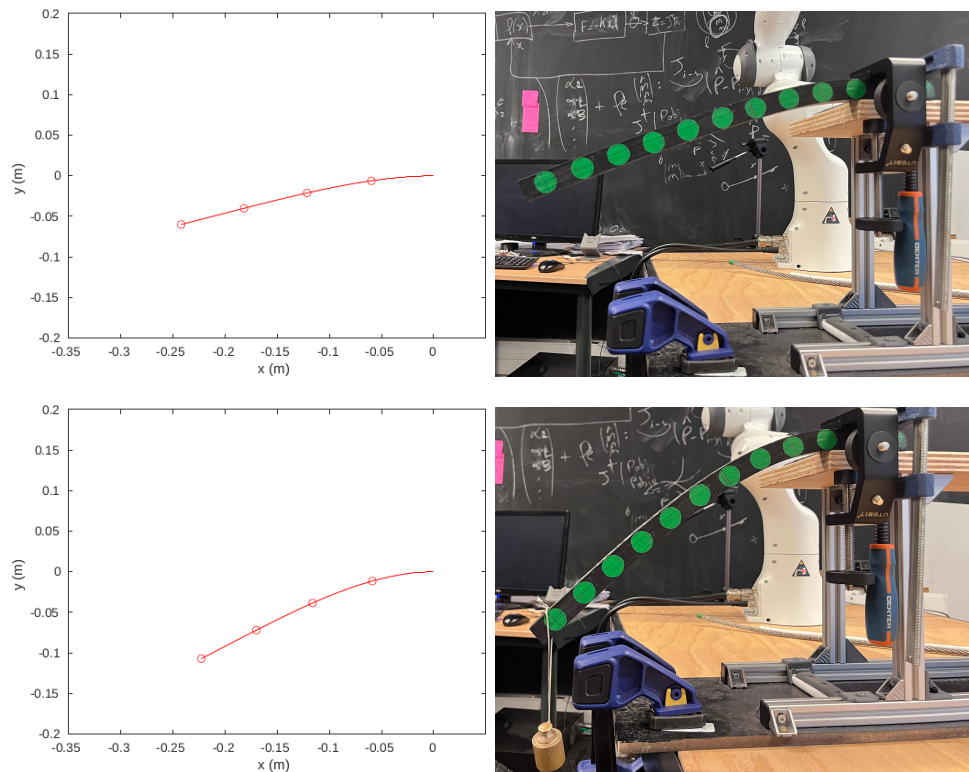
$$\text{subject to : } \left\{ \begin{array}{l} p(s=L) \subset Ws, \\ \dot{p} = Rv, \\ \dot{R} = R\hat{u}, \\ \dot{n} = -f, \\ \dot{m} = -\dot{p} \times n - l, \\ n(L) = 0, \\ m(L) = 0 \end{array} \right. \quad \begin{array}{l} \\ \\ \\ \\ \text{Conditions libres en } s = L \\ \end{array}$$

Pour davantage de précision, on relance l'expérience en attachant des poids connus à différents points de l'objet, E et ν sont obtenus en faisant la moyenne des résultats sur l'ensemble des essais (voir figure 3.13)

Pour valider l'approche, on compare la configuration de l'objet au cours du temps à p_{obj}^i . On définit l'erreur comme la distance moyenne entre l'objet et p_{obj}^i :

$$error = \frac{1}{n} \sum_{i=1}^n dist(\Lambda, p_{obj}^i) \quad (3.7)$$

Dans un premier temps, on teste l'algorithme pour cinq formes désirées, avec un nombre et un emplacement de points objectifs différents. N'importe quelle configuration de la bande peut être utilisée comme configuration initiale à condition que l'on connaisse le vecteur des valeurs initiales Γ_0 . La configuration au repos, droite, non déformée est utilisée comme configuration initiale de l'objet dans ce chapitre. On trace l'erreur en 3D et en 2D au cours du temps pendant la manipulation sur les figures 3.16. Ces erreurs correspondent aux formes finales représentées sur les figures 3.14 et 3.15. Pour chacune des expériences, on a une photo de la forme finale (à gauche) superposée à p_{obj}^i , et (à droite) une superposition entre la solution prédite par le modèle (rouge), la forme finale mesurée de l'objet (bleu) ainsi que p_{obj}^i (noir). On constate que dans tous les cas, l'algorithme d'optimisation nous fournit une solution qui

FIGURE 3.13 – Estimation de E et ν avec différentes charges

atteint bien p_{obj}^i . De plus, les solutions prédites par le modèle et les formes finales mesurées de l'objet sont également très similaires, ce qui signifie que les hypothèses du modèle sont respectées.

La distribution des erreurs finales pour les expériences conduites est représentée sur les figures 3.17a. L'erreur finale varie entre 1.6 mm et 3.6 mm en 3D. A cette échelle, l'erreur de la camera n'est pas négligeable, surtout pour les mesures de profondeur qui sont moins précises avec la Kinect V2 que dans les deux autres directions. On affiche alors également les mesures dans le plan en 2D sur les figures 3.17a et 3.16b.

3.4.3 Répétabilité

La répétabilité pour une application de ce type, et d'autant plus pour une application industrielle, est une caractéristique clé. Lors de la manipulation d'objets déformables, la répétabilité n'est jamais assurée. De nombreuses hypothèses posées sont rarement parfaitement vérifiées en réalité. On peut citer par exemple les hypothèses sur le comportement purement élastique des objets, le fait de négliger la viscosité, la courbure ou la configuration initiale

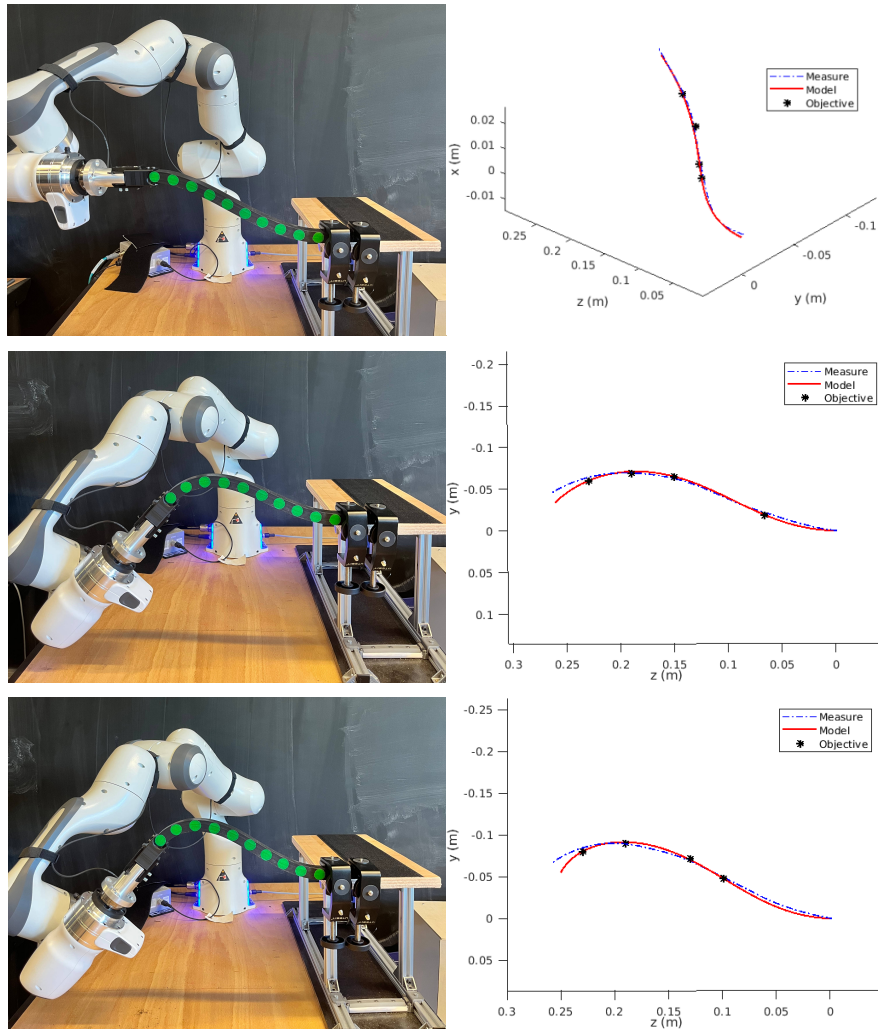


FIGURE 3.14

de l'objet, etc. De fait, même pour une résolution statique du problème, la forme finale peut ainsi varier en fonction du chemin suivi ou de l'état de l'objet à l'initialisation. En contrôlant entièrement la forme de l'objet durant sa trajectoire, on s'assure ainsi de la répétabilité de la méthode. Sur la figure 3.17b, on observe la variance de l'erreur finale en répétant la même tâche 8 fois successivement, avec les mêmes paramètres (même configuration initiale et p_{obj}^i). L'ensemble des 8 essais réalisés successivement produisent une erreur finale entre 1.5mm et 2.6mm. On constate de plus que la distribution des erreurs inter-tâche (figure 3.17a) est d'une échelle similaire à celle intra-tâche (figure 3.17b). Cela montre donc la consistance et la répétabilité de l'approche. Les seuls cas d'échec que nous avons eu sont lorsque l'algorithme ne trouve pas de solution pour des cas très complexes de déformations.

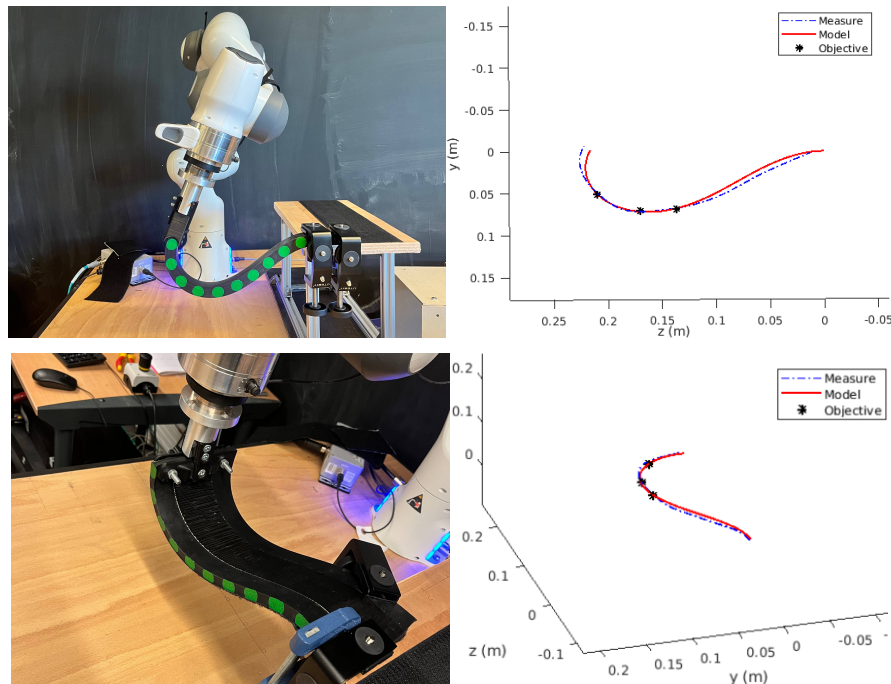


FIGURE 3.15

3.4.4 Modélisation d'objets planaires avec le modèle Cosserat

Bien que le modèle Cosserat soit un modèle d'objets linéaires, il peut dans une certaine mesure être étendu à d'autres objets avec des géométries proches. Plus l'objet est linéaire, c'est-à-dire qu'une dimension de l'objet est très grande devant les deux autres, plus le modèle est précis, mais cela ne signifie pas qu'il soit inapplicable à d'autres objets. Sur la figure 3.15, on peut voir qu'on a appliqué la méthode à une bande élastique plane (paramètres sur la table 3.1), c'est-à-dire qu'une dimension de l'objet est très petite devant les deux autres. L'approche fonctionne car la dimension de l'objet selon laquelle est appliquée la déformation respecte les hypothèses du modèle et est petite devant la longueur de l'objet. Dans ce cas spécifique, une déformation en trois dimensions n'est cependant pas possible. Le schéma sur la figure 3.18 illustre un cas d'échec de la généralisation pour un objet plane, tandis que le schéma en figure 3.19 illustre un cas de réussite. Il est important de noter que la rigidité de l'objet joue également un rôle très important dans la possibilité ou non d'appliquer le modèle Cosserat à un objet plane. Bien que les hypothèses de validité du modèle soient purement géométriques, un objet plane très souple est beaucoup plus facile à modéliser avec un modèle Cosserat qu'un objet plane rigide. L'objet rigide sera en effet beaucoup plus

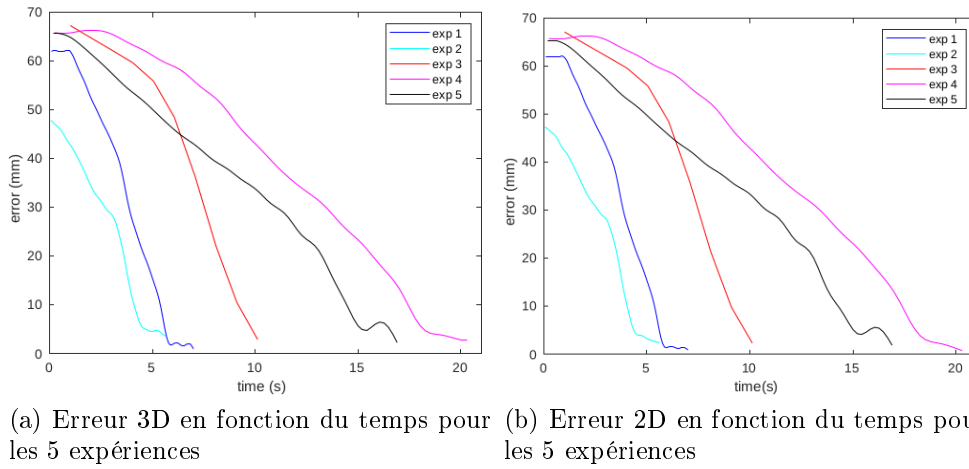


FIGURE 3.16

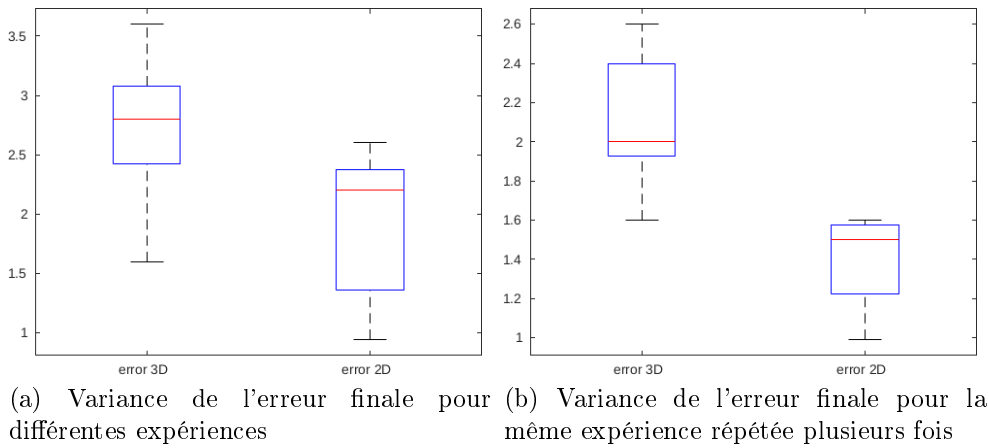


FIGURE 3.17

sensible aux erreurs du modèle car un léger écart en position ou orientation par rapport à la forme prédite engendrera sur le manipulateur des forces de rappel élastiques très élevées.

3.4.5 Discussion et conclusions

Dans ce chapitre, nous avons proposé une approche basée sur la théorie de Cosserat qui permet de réaliser des tâches de manipulation d'objets déformables. Nous avons dans un premier temps modifié la formulation classique posée originellement pour le contrôle de robots linéaires afin de l'adapter à la manipulation d'objets. Puis, dans un second temps, nous avons inclus le modèle de l'objet dans un problème d'optimisation afin de déterminer

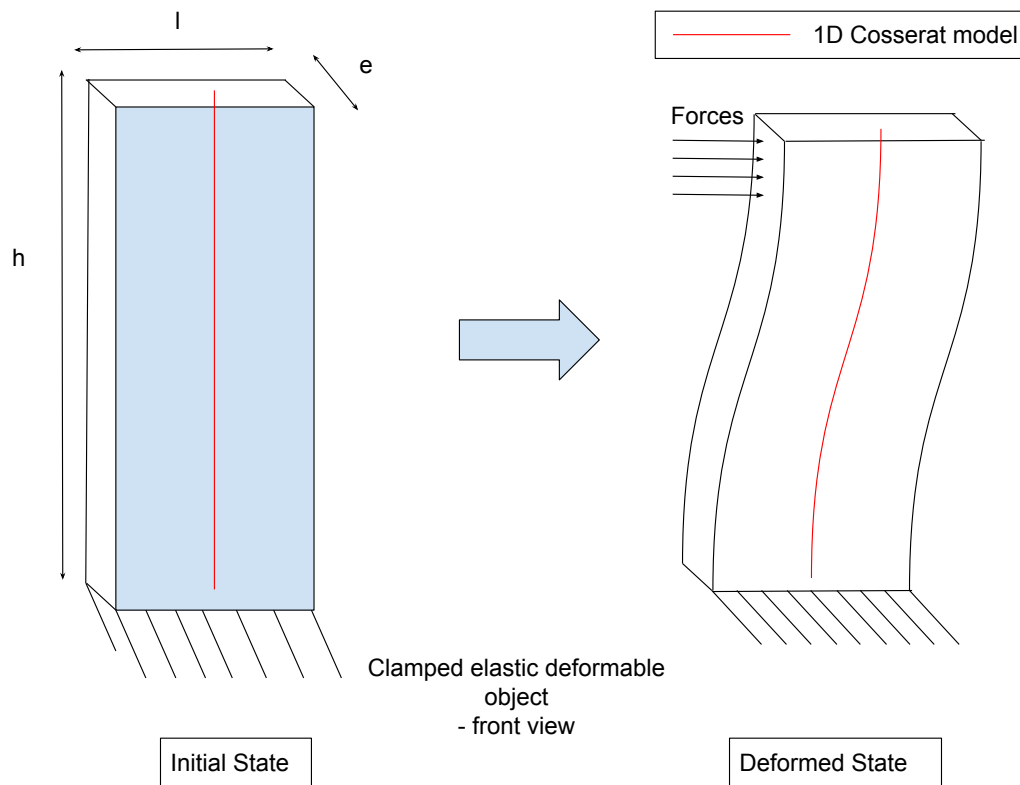


FIGURE 3.18 – Schéma représentant un cas d'échec pour la généralisation : déformation selon une dimension non négligeable par rapport à la longueur de l'objet

une configuration de l'objet qui minimise à la fois la distance par rapport à l'objectif, et également les efforts appliqués sur la bande. En discrétisant la position des points objectifs entre les états initiaux et finaux, nous avons résolu itérativement le problème. Cela nous permet non seulement d'améliorer la robustesse de l'algorithme et d'éviter les minima locaux, mais également de générer une trajectoire du robot qui minimise le critère d'optimisation en tout point de la trajectoire.

On a donc une méthode robuste, répétable et précise qui permet de générer une trajectoire pour déformer un objet tout en prenant en compte les contraintes appliquées par le robot. Toutefois la méthode a également des défauts :

- **Génération de trajectoire hors ligne** : Garantir la robustesse et le contrôle sur la trajectoire en résolvant itérativement le problème est très coûteux en calculs. Résoudre le modèle Cosserat inverse est déjà en soit un défi à intégrer dans une boucle de contrôle fermée, alors le résoudre itérativement pour chaque configuration intermédiaire

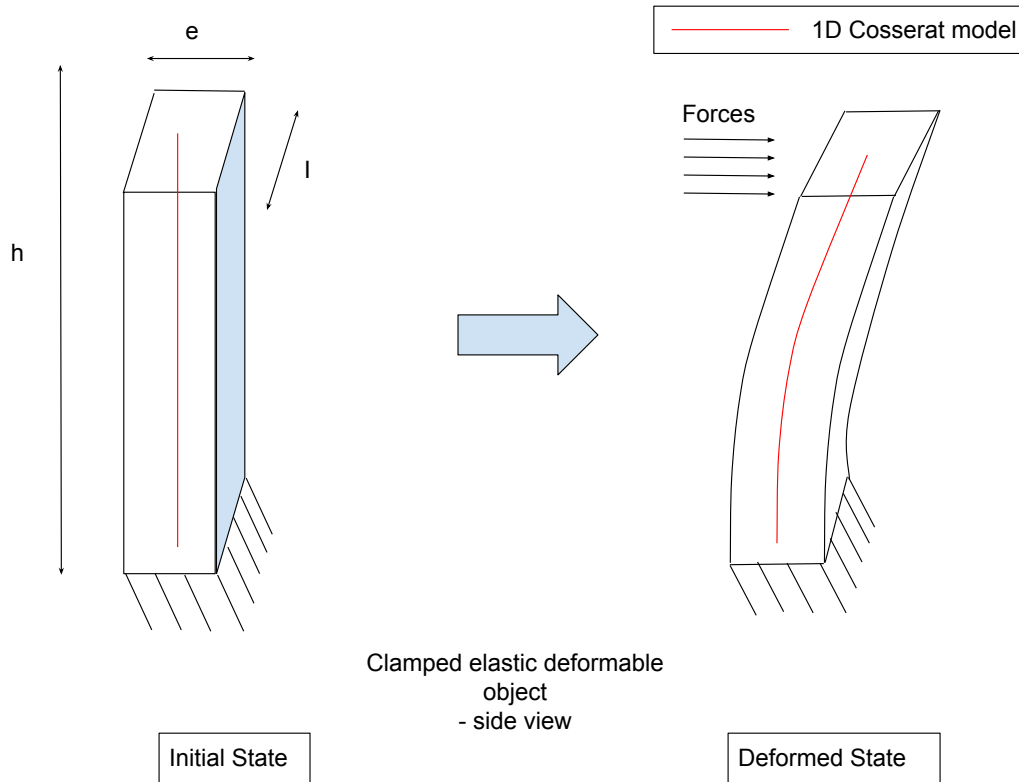


FIGURE 3.19 – Schéma représentant un cas de réussite pour la généralisation : déformation selon une dimension négligeable par rapport à la longueur de l'objet

doit être fait hors-ligne. Cela résulte en une erreur statique en fin de manipulation qui est liée à de multiples erreurs : estimations des paramètres de l'objet, erreurs sur les conditions initiales de l'expérience, hypothèses du modèle non respectées. De plus, la trajectoire n'étant jamais recalculée au cours de la tâche, la méthode ne permet pas de corriger les erreurs en temps réel. Le temps de calcul moyen pour chaque configuration dépend très fortement des paramètres de l'objet, de la complexité de la tâche et varie entre 0.5 secondes pour les problèmes les plus simples à 2.5 secondes pour les problèmes les plus complexes. Ces temps de calcul dépendent de notre setup : CPU Intel i7 et la puce graphique standard, sans parallélisation.

- **Sensibilité à l'estimation des paramètres élastiques** : une des principales sources d'erreur statique, et l'un des principaux défauts communs aux modèles mécaniques physiquement réalistes. L'estimation de ces paramètres joue beaucoup dans la précision du modèle car ce sont ceux-ci, ainsi que la géométrie de l'objet qui

vont définir le comportement de l'objet soumis à des déformations. Les deux principaux paramètres sont le module d'Young E , et le coefficient de Poisson ν . Bien que la géométrie de l'objet soit souvent facile à déterminer avec un mètre, ces paramètres sont notoirement difficiles à exprimer précisément. De plus, cela suppose que l'objet est composé d'un matériau homogène et isotrope, ce qui exclut de nombreux objets comme les câbles en fils de métal tressé, ou encore tous les matériaux composites. Dans ces cas, il est nécessaire de déterminer des constantes équivalentes, qui sont souvent moins précises car elles peuvent changer en fonction de la configuration de l'objet, ou encore avoir des comportements complètement différents d'un matériau homogène isotrope (par exemple un coefficient de Poisson négatif).

Approche globale et robustesse de la méthode

Dans le chapitre précédent, nous avons présenté notre approche permettant de générer une trajectoire pour réaliser une tâche de manipulation d'objet linéaire, un cas d'étude classique de contrôle de forme. Ce cas d'étude est une tâche de référence dans la communauté de la manipulation robotique d'objets déformables. Cependant, il s'agit d'une tâche générale qui est rarement appliquée en l'état dans l'industrie.

Dans ce chapitre, nous allons proposer dans un premier temps d'utiliser à la place de l'algorithme d'optimisation local, un algorithme global. On s'assure ainsi d'éviter totalement les minima locaux. De plus, on sépare les objectifs en position et en énergie en deux objectifs séparés. On résout donc un problème d'optimisation multi-objectif. On génère ainsi le front de Pareto [Van Veldhuizen 1998] du problème de sorte à visualiser l'ensemble des optimums de Pareto. Choisir une solution adaptée à la tâche est ainsi plus intuitif.

Dans un second temps, on va appliquer notre approche à un cas plus concret d'application industrielle. Plus spécifiquement, on va se pencher sur une tâche de détachement d'un objet planaire, similaire au démoulage de semelle du cas d'étude Plastinher. Il est difficile de répliquer exactement un environnement industriel en laboratoire. On va donc proposer une maquette équivalente, présentant les mêmes difficultés que la tâche réelle, mais qui soit applicable dans un laboratoire. Pour cela, on remplace la semelle insérée dans le moule par une bande élastique attachée à l'aide d'un velcro à une table. On va ainsi modéliser les efforts appliqués par le velcro à l'objet afin de générer une trajectoire optimale du robot pour détacher la bande.

Enfin, on va rendre la tâche plus robuste à un ensemble de paramètres qu'il peut être difficile de contrôler hors d'un laboratoire. Pour cela, on va introduire de la compliance dans la tâche en remplaçant le contrôleur en position du robot par un contrôleur en impédance. En rendant le préhenseur plus compliant, on permet ainsi à l'objet, compliant lui même, de guider le préhenseur durant la tâche. L'objet tend à se déformer vers une configuration qui minimise l'énergie potentielle élastique. L'objectif est ainsi de laisser le préhenseur être guidé par

l'objet dans les directions non nécessaires à la tâche tout en étant rigide dans les directions qui le sont.

4.1 Problème multiobjectif et recherche globale

Jusqu'ici, nous avons formulé notre fonction objectif avec un système de poids sur chaque objectif (erreur de distance et minimisation de l'énergie). Toutefois, ces poids n'ont pas de signification physique évidente. En plus d'être peu intuitifs d'un point de vue physique, ces poids peuvent également être difficiles à déterminer de façon à ce que l'algorithme réalise bien ce qu'on attend de lui pour une tâche donnée.

De plus, bien que nous ayons fortement réduit la probabilité de tomber dans des minimas locaux en résolvant chaque étape du problème de manière séquentielle, nous n'avons pas de garantie qu'il soit impossible de se retrouver dans un minimum local à un point donné. On sait uniquement que plus le point d'initialisation fourni à l'algorithme est proche de la solution globale, plus la probabilité d'obtenir une solution globale est élevée. De fait, on peut raisonnablement supposer que lorsque la différence entre les solutions à chaque étape successive tend vers zéro, alors la probabilité de tomber dans un minimum local tend également vers zéro. En d'autres termes, cela signifie qu'il suffit de donner à l'algorithme un point d'initialisation qui se situe dans le voisinage convexe de la fonction de coût autour de la solution désirée. Si l'on connaît ce voisinage, on peut de fait définir les étapes intermédiaires du problème de manière à ce que la solution de chaque étape se trouve dans le voisinage de l'étape suivante. Ainsi, on est assuré d'obtenir une solution globale à chaque étape.

Cependant, il est très difficile d'estimer analytiquement ce voisinage, d'autant plus qu'il n'est pas constant à chaque étape du problème. Jusqu'ici la solution était de déterminer expérimentalement ce voisinage en résolvant le problème et en ajustant la discrétisation de l'objectif jusqu'à éviter les minimas locaux. Bien que cela fonctionne en pratique pour un problème donné, il est nécessaire de répéter cette étape à chaque fois que l'on cherche un nouveau problème.

Comme solution à ces deux difficultés, on propose d'utiliser un solveur global (algorithme génétique), et de reformuler le problème d'optimisation de sorte à faire apparaître les deux objectifs. On obtient alors le problème multi-objectif suivant (voir eq. 4.1) :

$$\begin{aligned}
\min_{n(0), m(0)} c_1 &= \frac{1}{2} \int_0^L (v(s) - \tilde{v}(s))^T K_{tra}(s) (v(s) - \tilde{v}(s)) \\
&\quad + (u(s) - \tilde{u}(s))^T K_{rot}(s) (u(s) - \tilde{u}(s)) ds \\
c_2 &= \sum_{i=1}^k dist(\Gamma, p^{obj}(s_i)),
\end{aligned} \tag{4.1}$$

sujet à :

$$\left\{ \begin{array}{l} p(s=L) \subset Ws, \\ \dot{p} = Rv, \\ \dot{R} = R\hat{u}, \\ \dot{n} = -f, \\ \dot{m} = -\dot{p} \times n - l \end{array} \right.$$

Au lieu du système de poids utilisé précédemment, on peut ainsi choisir une solution sur le front de Pareto obtenu à partir de l'algorithme génétique. On peut ainsi hiérarchiser les objectifs en choisissant une solution sur le front de Pareto qui privilégie le respect d'un objectif par rapport à l'autre. Le désavantage de cette approche par rapport à la précédente est le temps d'exécution de l'algorithme de recherche global qui est beaucoup plus long que celui d'un algorithme basé sur la méthode des points intérieurs. Toutefois, pour une planification hors-ligne, le temps de calcul n'est pas un problème. Des exemples de fronts de Pareto pour différents objectifs, obtenus à l'aide de l'algorithme génétique de Matlab sont présentés en figure 4.1.

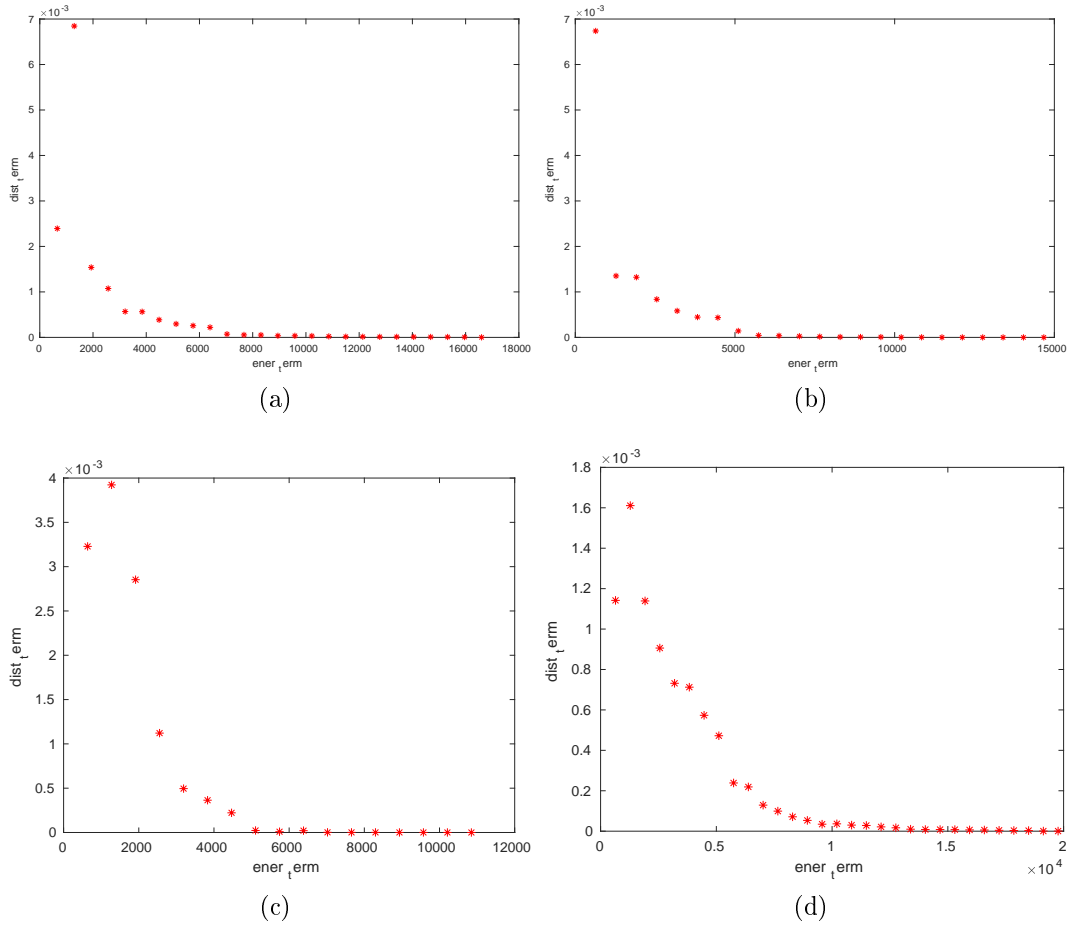


FIGURE 4.1 – Fronts de Pareto pour différentes formes désirées

4.2 Application à la tâche de détachement d'un objet planaire

4.2.1 Introduction et objectifs de la partie

Dans la partie précédente, nous avons présenté un algorithme permettant de générer une trajectoire de l'organe terminal du robot permettant de déformer un objet élastique linéaire de manière optimale. Le problème d'optimisation prend en compte deux fonctions de coût, la première minimise l'énergie potentielle élastique dans l'objet, et la seconde la distance entre l'objet et une configuration cible que l'on cherche à atteindre. Cette configuration cible est ensuite projetée spatialement sur la configuration initiale de l'objet afin de générer un ensemble de configurations intermédiaires désirées de l'objet. En résolvant le problème d'optimisation pour chacune de

4.2. Application à la tâche de détachement d'un objet planaire 67

ces configurations intermédiaires, on obtient ainsi une trajectoire du robot permettant de minimiser les objectifs tout en respectant les contraintes à chaque étape de la trajectoire. Nous avons testé la méthode avec une application classique de contrôle de forme d'un objet linéaire déformable à l'aide d'un bras Franka Emika.

Le but de cette section est d'appliquer cette méthode à un cas industriel de manipulation robotique d'objets déformables. Comme exemple, nous allons tenter de résoudre dans cette partie le cas d'étude du démoulage de semelle proposé par le projet SOFTMANBOT. On peut voir des illustrations de différentes semelles dans leurs moules sur la figure 4.2, et du processus de démoulage sur la figure 4.3.

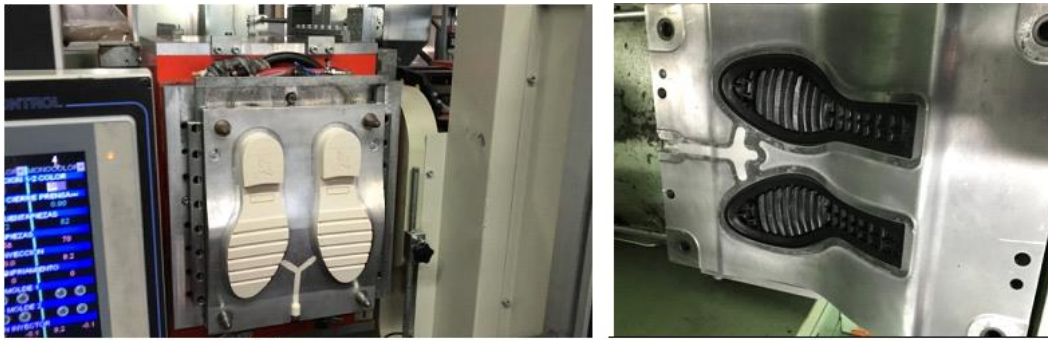


FIGURE 4.2 – Semelles de caoutchouc dans le moule

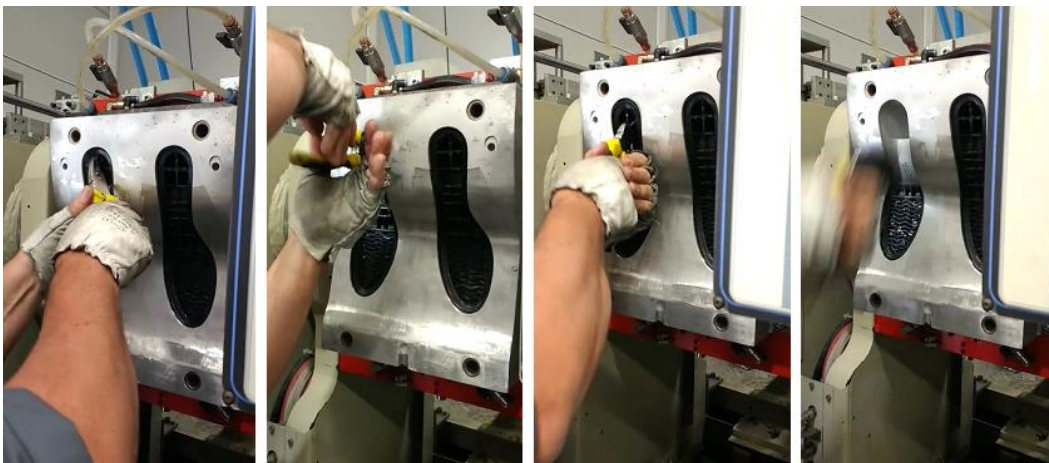


FIGURE 4.3 – Processus manuel de démoulage de semelle

Nous ne possédons pas au laboratoire l'infrastructure exacte (moules, semelles à la bonne température, etc), nous avons donc tenté de créer une maquette, plus facile à mettre en place au laboratoire que l'équipement

industriel, mais qui conserve les mêmes difficultés qui rendent la tâche de démoulage de semelle difficile à résoudre. Ces difficultés sont :

- Contacts complexes et interactions très difficiles à modéliser entre la semelle et le moule.
- Manipulation d'une semelle déformable en caoutchouc chaud.
- Semelle difficile à extraire, même pour un humain non entraîné : la trajectoire et la direction des efforts à appliquer ne sont pas triviaux et appliquer des efforts importants sans la bonne trajectoire ou dans une direction non optimale ne suffit pas à réaliser la tâche.
- Nécessité de contrôler les forces à appliquer. Le matériau est fragile et l'intervalle de force dans lequel il a un comportement élastique est petit, les déformations plastiques étant à éviter.

4.2.2 Présentation de la maquette

Dans cette partie, nous reprenons le dispositif présenté dans la partie précédente avec le bras Franka Emika et la kinect V2 pour la vision 3D. En plus de cela, nous ajoutons un capteur ATI 6 axes de forces/moments sur le bras, de sorte à mesurer les efforts déployés au point de contact entre le préhenseur et l'objet manipulé. Le capteur, monté sur le robot est représenté sur la figure 4.4. On applique un filtre Gaussien sur les données obtenues afin de se débarrasser des bruits de mesure.

Pour la maquette représentant le démoulage d'une semelle élastique, la semelle en elle-même est représentée par une bande élastique rectangulaire en caoutchouc, saisie à une extrémité par le robot. On utilise ainsi la bande élastique en deux dimensions utilisée dans le chapitre précédent. Les paramètres que l'on a estimé pour modéliser l'objet sont représentés sur le tableau 3.1.

La maquette pour le moule est plus complexe, les interactions et frottements statiques sont difficiles à modéliser et il n'est pas évident de trouver un équivalent fidèle à mettre en place. Nous avons choisi d'attacher la bande à la table à l'aide d'un velcro adhésif. Bien que les deux interactions soient de nature différentes, elles partagent de nombreux points communs :

- Les deux sont difficiles à modéliser précisément.
- La dynamique du démoulage et du détachement est semblable : plus l'objet se détache/démoule, plus le bras de levier est important et moins les efforts nécessaires pour le démoulage/détachage sont importants.
- Les efforts à appliquer dans les deux cas sont importants et une trajectoire non optimale peut faire échouer la manipulation.

Le dispositif final est représenté sur la figure 4.5.



FIGURE 4.4 – Capteur de force fixé au poignet du robot qui mesure les forces et moments appliqués au point de saisie

4.2.3 Modélisation

Comme évoqué précédemment, les interactions entre la bande élastique et le velcro sont complexes à modéliser. Pour obtenir le comportement le plus fidèle possible, on distingue sur la bande trois sections distinctes qui vont être intégrées avec des conditions aux limites et des contraintes différentes (voir figure 4.6) :

- La première partie $[0; l_a]$, complètement contrainte et attachée à la table par le velcro.
- La seconde partie $[l_a; l_{pa}]$, partiellement attachée et en cours d'arrachage. Des forces réparties dues aux efforts du velcro s'appliquent à la bande.
- la dernière partie $[l_{pa}; L]$, déjà détachée, aucune contrainte ne s'applique sur la bande autre que celles du robot.

Comme vu dans le chapitre précédent, l'objectif final de la position de la forme souhaitée est discrétisé afin de former l'ensemble des objectifs intermédiaires. On pose d_{detach} (voir figure 4.6), la distance entre la table et l'objet à partir de laquelle on considère une section de la bande entièrement détachée. Les objectifs intermédiaires vont ainsi se déplacer de $l = L$ à $l = 0$ de sorte à détacher progressivement la bande. On peut voir les étapes et les

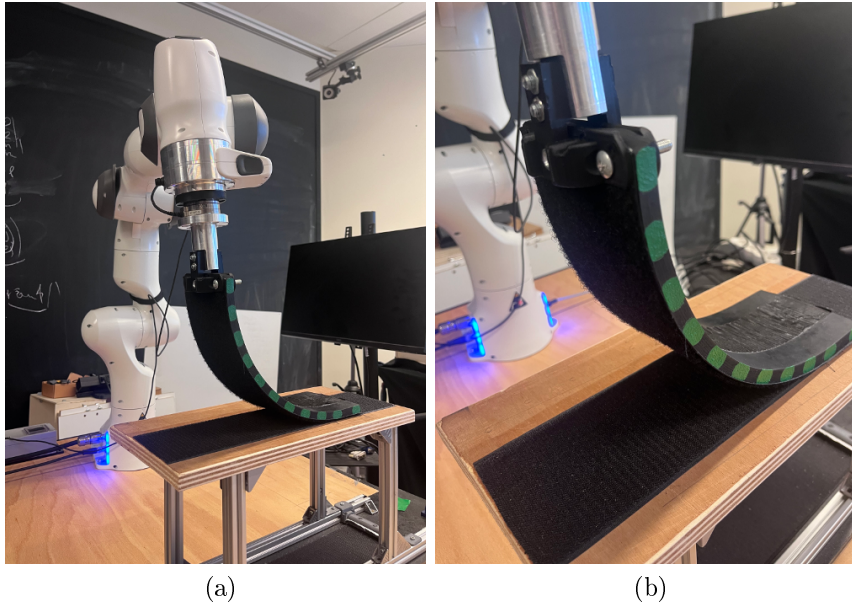


FIGURE 4.5 – Setup expérimental représentant la bande élastique partiellement attachée

configurations successives de la bande schématisées sur les figures 4.7 et 4.8b, les objectifs intermédiaires en position p_i^{detach} y sont également représentés.

On n'utilise pas de modèle complexe d'adhésion [Autumn 2006] pour identifier les efforts appliqués par le velcro sur la bande. En plus d'être difficilement intégrables à notre approche, ces modèles sont spécifiques à certains types d'interactions et non généralisables au cas d'étude réel du démoulage.

On a choisi à la place, d'estimer ces efforts expérimentalement. Ainsi, pour appliquer notre méthode à d'autres tâches, il suffira de procéder à une nouvelle estimation plutôt que d'intégrer un autre modèle de friction ou d'adhésion. Cette approche a l'avantage d'être plus facile à généraliser au détriment d'une moins bonne précision.

Pour cela, on attache l'organe terminal au velcro à l'aide d'un tampon imprimé en 3D. On applique un déplacement selon l'axe normal à la bande de velcro. On mesure ainsi avec le capteur de force les efforts nécessaires pour détacher l'organe terminal (voir figure 4.9).

En utilisant le mode de guidage manuel du robot Franka Emika et en procédant lentement de sorte à minimiser les effets dynamiques, on arrache le tampon. On mesure ainsi à l'aide du capteur de force le profil des efforts nécessaires pour détacher le tampon. On entre ensuite ce profil dans le modèle en tant qu'efforts extérieurs pour la phase partiellement attachée.

4.2. Application à la tâche de détachement d'un objet planaire 71

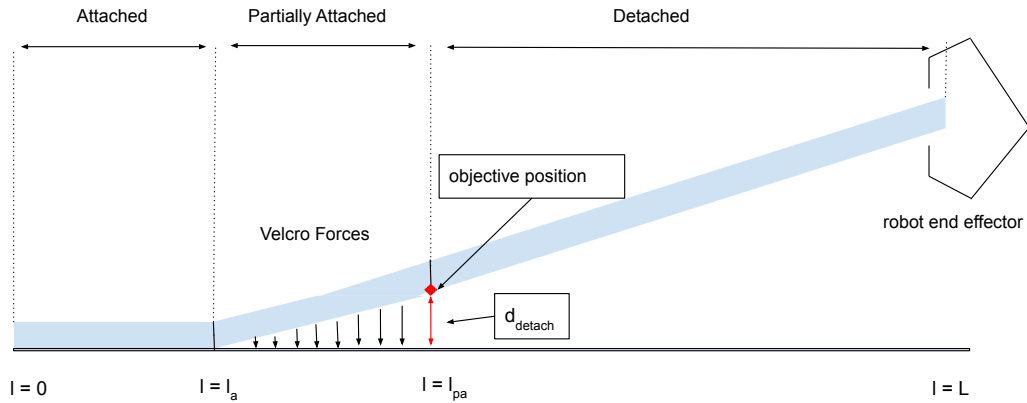


FIGURE 4.6 – Trois parties de la bande représentant les trois boucles d'intégration

En terme d'intégration des équations de Cosserat dans la bande, on a ainsi nos trois parties :

- La partie attachée qui n'est pas intégrée puisque statique
- La partie partiellement attachée qui est soumise aux efforts estimés précédemment
- La partie complètement détachée

En intégrant sur la partie partiellement attachée, on obtient donc les conditions initiales pour la partie détachée. Bien que l'on puisse supposer à raison que les erreurs de modélisation des efforts appliqués par le velcro peuvent être importantes, on peut toutefois noter que la partie partiellement attachée est en réalité de l'ordre du millimètre. Même si l'erreur relative sur la configuration de cette portion peut être élevée, l'erreur absolue par rapport à la bande entière est relativement faible. De plus, de par sa petite taille et le matériau utilisé, la raideur effective sur cette portion de la bande est extrêmement élevée, l'erreur sur les efforts extérieurs doit donc être très élevée pour donner une erreur en position significative. On peut donc supposer que ces erreurs de modélisation ont un impact faible sur la configuration finale de l'objet.

4.2.4 Contrôle en impédance

Nous avons montré dans le chapitre précédent que notre approche permettait un contrôle précis de la forme à l'aide du modèle Cosserat. Toutefois, aussi précis que soit le modèle Cosserat, il reste nécessairement des erreurs, qu'elles soient liées au modèle en lui-même, à l'estimation des constantes élastiques ou de tout autre paramètre mesuré au préalable. La modélisation des efforts extérieurs de contact est également une source

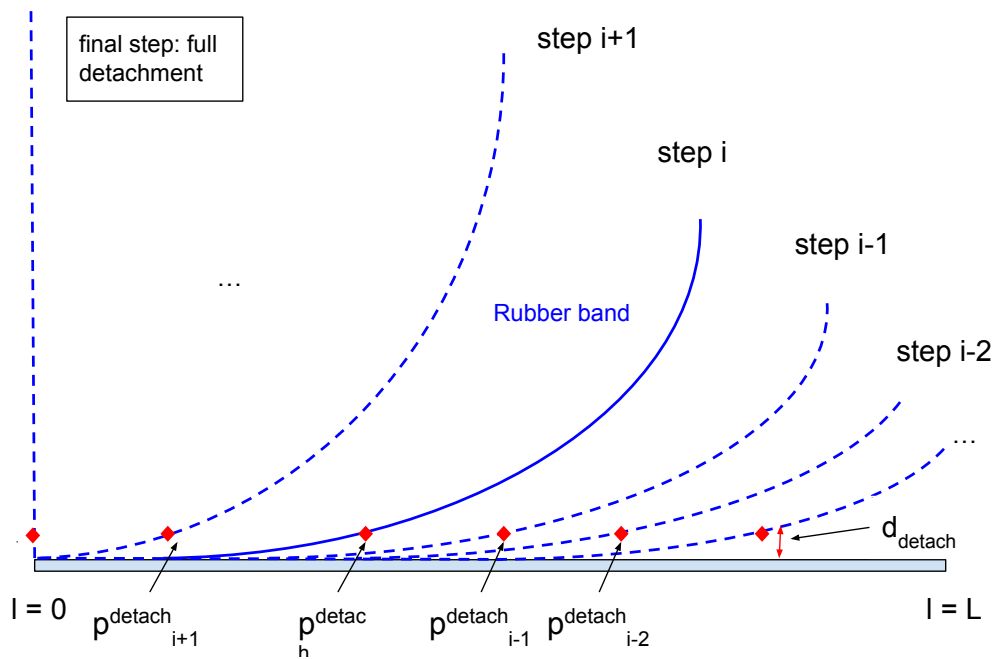


FIGURE 4.7 – Étapes du détachement et objectifs intermédiaires associés

d'erreur. Même si nous avons réduit l'impact de ces erreurs à la fois en intégrant indépendamment sur différentes parties de la bande et en utilisant la position plutôt que les efforts comme commande, on peut tout de même s'attendre à un écart entre la configuration $i-1$ du modèle et la configuration réelle de l'objet.

D'autres sources d'erreurs particulièrement importantes sont liées aux conditions initiales de l'expérience (position de la bande, déformation résiduelle de l'objet, etc). Ces conditions initiales sont rentrées sous forme de paramètres dans le modèle. Par exemple, un très léger défaut d'alignement de la bande élastique par rapport à l'axe de référence entraîne sur le robot des forces de rappel importantes dans les plans parallèles à la bande. Cela est dû à la rigidité du matériau et à la géométrie de l'objet. Ainsi, une légère erreur d'orientation de quelques degrés de la bande dans son état initial peut faire échouer la tâche. Plus précisément, l'orientation de la bande et la trajectoire du robot ne sont plus alignées, le robot exerce donc des efforts dans la direction de la tranche de l'objet (axe x sur la figure 4.10). Le phénomène est illustré sur la figure 4.10. Les forces de rappel de la bande peuvent devenir importantes au point que le robot ne peut plus déployer assez de force pour arracher la bande.

Afin de compenser ces erreurs dues aux phénomènes non ou mal modélisés, nous nous sommes penchés sur le contrôleur du robot. Nous aurions pu tenter

4.2. Application à la tâche de détachement d'un objet planaire 73

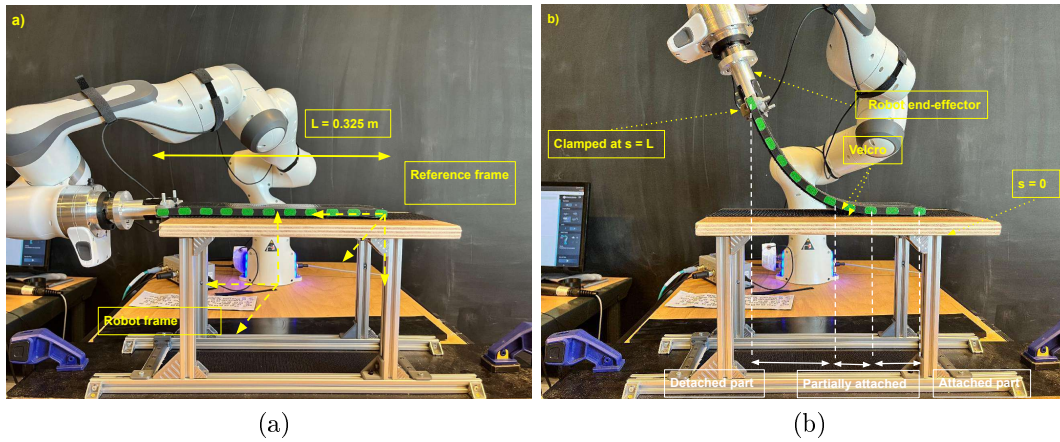


FIGURE 4.8 – a) Dispositif expérimental représentant la bande à l'initialisation.

b) Dispositif expérimental au cours du détachement.

d'implémenter un contrôleur avec une boucle fermée sur la configuration de l'objet. Cependant, même si l'on met de côté la difficulté d'inclure un modèle aussi coûteux en calcul dans une boucle temps réel, cela nous aurait uniquement permis de réduire l'erreur entre la configuration du modèle et la configuration réelle de l'objet. Or, d'une part on cherche à obtenir également une configuration optimale par rapport aux efforts appliqués sur l'objet ; le contrôle de forme n'est pas notre unique objectif. D'autre part, le modèle étant intégré à la boucle fermée, s'il est partiellement faux, la forme de l'objet va converger vers une configuration non optimale de l'objet.

La solution que nous proposons est donc d'ajouter de la compliance dans la manipulation avec un contrôleur en impédance. L'objet étant lui même compliant, la dynamique de l'objet durant la manipulation va guider le manipulateur vers une position d'équilibre stable, soit une configuration qui minimise l'énergie potentielle élastique dans l'objet. Si l'on définit bien les paramètres de raideur et d'amortissement de notre contrôleur, on aurait alors un manipulateur assez rigide pour réaliser la tâche, tout en étant à la fois assez compliant pour compenser les erreurs du modèle et minimiser réellement les efforts appliqués. Le manipulateur doit donc être plus raide dans le plan de la tâche (plan sagittal, normal à la bande et qui passe par le centre de celle-ci dans son état initial), que dans les plans coronaux, dans lesquels on peut laisser l'objet guider l'organe terminal du robot.

Ainsi, l'algorithme génère une trajectoire optimale par rapport aux paramètres fournis dans le modèle, et le contrôleur du robot permet à la fois de compenser les erreurs de modélisation, et également de rendre la méthode

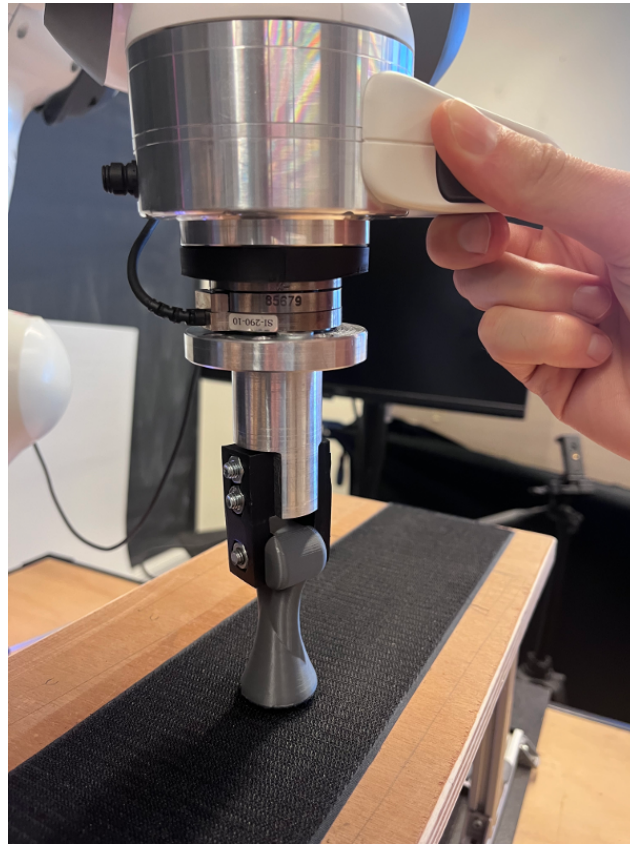


FIGURE 4.9 – Mesures des forces à l’aide du tampon et du capteur de forces/moments

robuste aux incertitudes sur les conditions initiales de l’expérience.

4.2.5 Détachement de la bande élastique

Dans un premier temps, on génère la trajectoire de la bande à l’aide de la méthode présentée précédemment. Cette trajectoire est exposée sur la figure 4.11.

De cette trajectoire de l’objet, on déduit la trajectoire de l’effecteur et on l’applique au système réel. Pour satisfaire les conditions aux limites, on suppose qu’un point de l’objet est toujours en contact avec la table. On rajoute donc manuellement une dernière étape afin de s’assurer du détachement total de l’objet.

Il est difficile d’avoir une mesure quantitative lorsque notre but est de réaliser une tâche spécifique et non de mesurer une erreur en position. Toutefois, pour montrer l’intérêt du contrôleur en impédance, on exécute deux fois la manipulation avec la même trajectoire et les mêmes conditions initiales

4.2. Application à la tâche de détachement d'un objet planaire 75



(a) Trajectoire du préhenseur et bande parfaitement alignées (b) Léger désalignement (c) Forces de rappel F exercées sur le préhenseur durant la manipulation

FIGURE 4.10 – Comparaison entre une bande alignée en accord avec le modèle (rouge) et une bande légèrement désalignée (bleu). Afin de suivre sa trajectoire, le robot doit appliquer des efforts supplémentaires dus aux forces de rappel de la bande.

de l'objet. Dans un premier temps, on utilise le contrôleur en impédance et dans un second temps le contrôleur en position. Pour ces deux essais, on enregistre à l'aide du capteur les forces et moments au point de saisie. Sur les figures 4.13a et 4.13b, on affiche les forces et moments selon chaque axe pour un détachement avec le contrôleur en impédance, et un détachement avec le contrôleur en position. Sur les figures 4.13c et 4.13d, on affiche et on compare la norme des forces et moments pour les deux expériences successives. On remarque que les forces et moments appliquées par le préhenseur sur la bande, et donc l'énergie élastique dans l'objet sont largement supérieurs lorsqu'on utilise le contrôleur en position plutôt que le contrôleur en impédance. Cela est en accord avec les hypothèses que l'on a faites sur les forces de rappel élastiques dues à un alignement initial de la bande qui ne correspond pas à celui entré dans le modèle. On observe bien sur la figure 4.13a que la plus grande diminution est selon l'axe x (voir figure 4.10). L'axe x correspond en effet à l'axe sur lequel les forces de rappel élastiques sont les plus élevées en raison de la géométrie de l'objet.

On remarque sur la figure 4.13b que les moments selon l'axe z sont équivalents sur les deux expériences. On a en effet constaté que le préhenseur était beaucoup moins compliant pour la rotation autour de son axe. Il semble que ce soit un problème matériel plus qu'informatique, on ne sait pas si cela est dû à une usure autour de cet axe ou au design du robot.

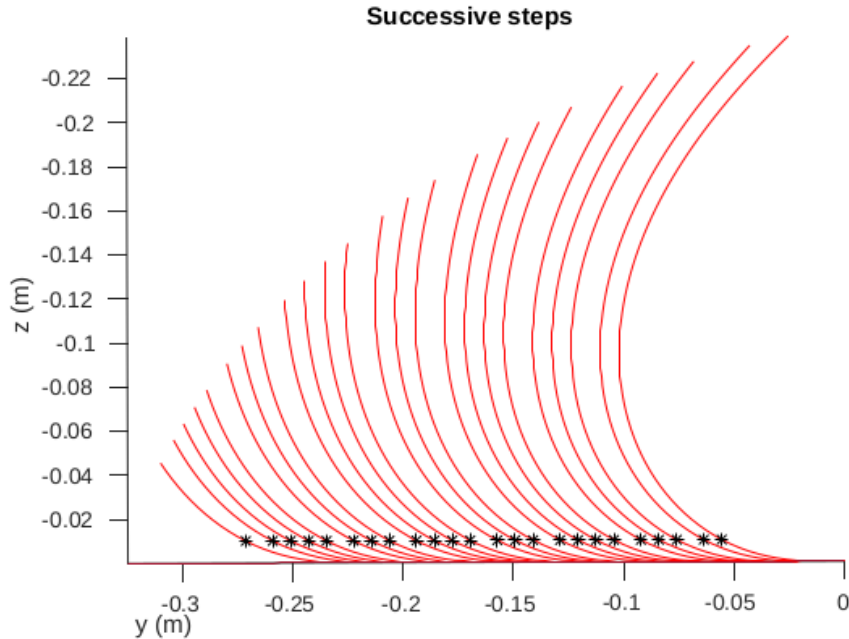


FIGURE 4.11 – Trajectoire générée de l’objet (rouge) durant la manipulation et objectifs (*) dans le plan de la tâche

4.3 Conclusion

Dans ce chapitre, nous avons dans un premier temps rendu l’approche robuste aux minima locaux en remplaçant l’algorithme d’optimisation local par un algorithme global. L’algorithme génétique utilisé permet ainsi de visualiser le front de Pareto du problème. Il est ainsi plus aisé de choisir une solution adaptée au problème que l’on cherche à résoudre. Dans les cas où la minimisation de l’énergie est primordiale, on peut ainsi choisir une solution moins précise en terme de position, mais qui résulte en une énergie plus faible dans l’objet. Inversement, pour une application où la précision est primordiale, on choisira une solution qui va privilégier la minimisation de l’erreur en position quitte à avoir une énergie plus élevée dans l’objet.

Dans un second temps, nous avons appliqué l’approche à une tâche spécifique de manipulation impliquant des contacts entre l’objet manipulé et l’environnement extérieur. Nous avons ainsi mis en place un problème similaire au démoulage de semelle du cas d’étude Plastinher. Pour remplacer l’interaction entre le caoutchouc et le moule, on a utilisé du velcro pour attacher une bande élastique à une table. On a ensuite généré une trajectoire du robot pour détacher la bande.

Afin de rendre l’approche plus compliante et plus robuste à l’ensemble des phénomènes non modélisés ou aux erreurs de modélisation, nous avons

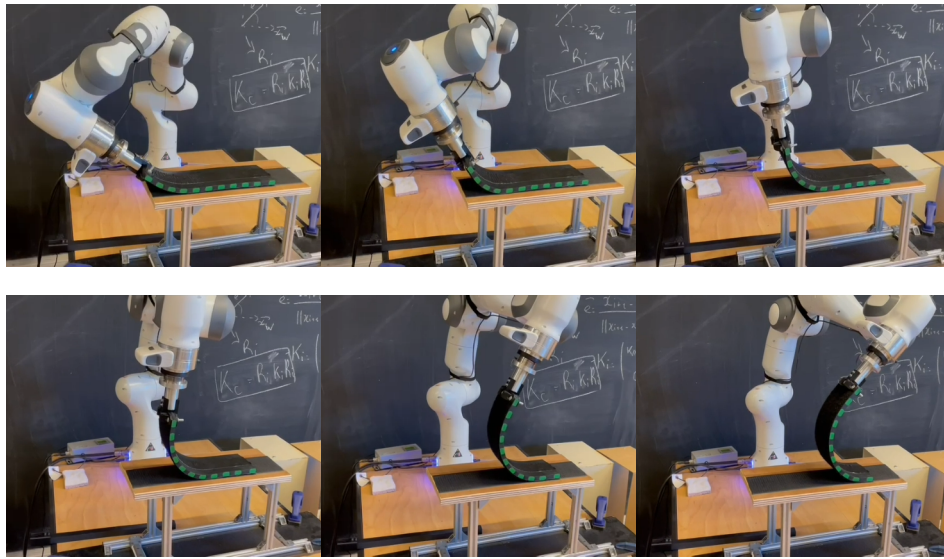
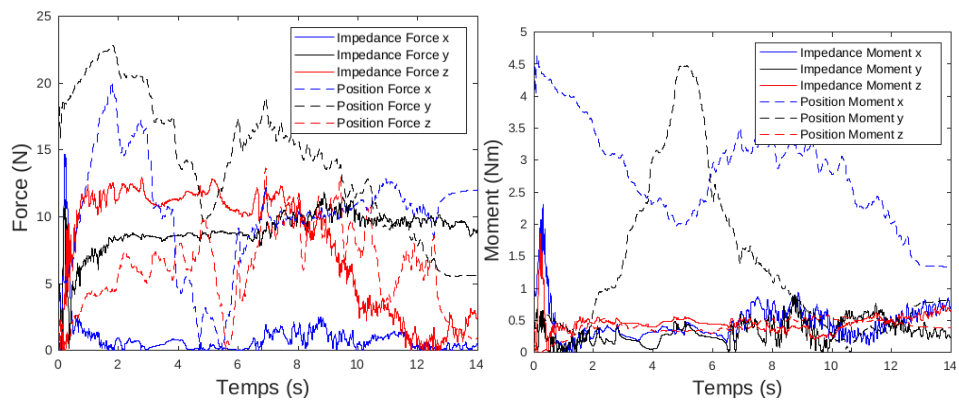
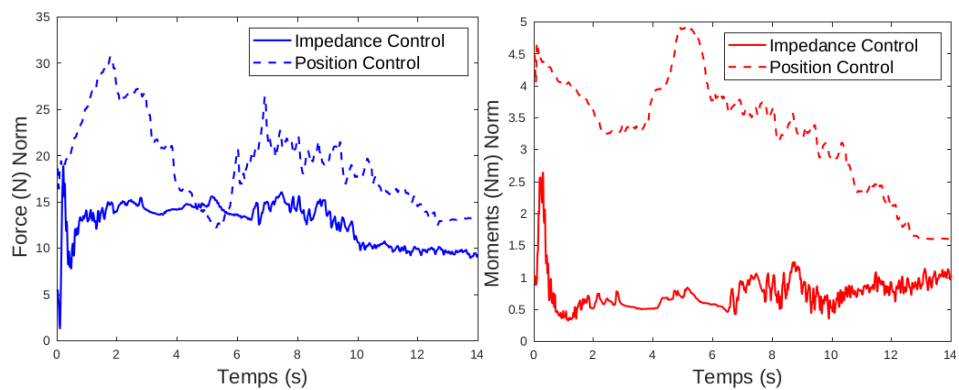


FIGURE 4.12 – Étapes successives du détachement de la semelle

remplacé le contrôleur en position du robot par un contrôleur en impédance. On a successivement réalisé deux expériences de détachement, l'une avec le contrôleur en impédance et l'autre avec le contrôleur en position. On a constaté que le contrôleur en impédance permet bien de réduire les efforts appliqués sur l'objet, et par extension l'énergie élastique dans l'objet.



(a) Mesures des forces au cours des deux manipulations (b) Mesures des moments au cours des deux manipulations



(c) Comparaison de la norme des forces au cours des deux manipulations (d) Comparaison de la norme des moments au cours des deux manipulations

FIGURE 4.13

Contrôle de la déformation en boucle fermée

Dans les chapitres 3 et 4, nous avons proposé notre méthode pour générer une trajectoire permettant de réaliser une tâche de manipulation d'objets linéaires tout en minimisant l'énergie élastique dans l'objet. Bien que cette méthode réponde à des problèmes concrets qui n'étaient jusqu'ici pas traités dans l'état de l'art tels que la résolution d'une tâche de manipulation complexe en contexte industriel, elle a toutefois des limites :

- **Trajectoire en boucle ouverte** : Trouver une trajectoire optimale en tout point tout en prenant en compte l'ensemble des paramètres de la tâche requiert un temps de calcul qui ne permet pas de recalculer la trajectoire en temps réel. Nous n'avons donc aucun moyen de corriger la trajectoire pendant la manipulation. De plus, les défauts du modèle ne peuvent pas être compensés et la forme de l'objet convergera toujours avec une erreur statique plus ou moins élevée.
- **Sensibilité à l'estimation des constantes élastiques** : Comme pour l'ensemble des modèles mécaniques, la précision du modèle dépend fortement de la précision des paramètres élastiques de l'objet. Ces paramètres sont notoirement difficiles à modéliser, et n'ont de sens physique que pour des objets homogènes et isotropes. Les objets non isotropes et matériaux composites ne sont généralement pas traités avec ces modèles.

Résoudre ces problèmes nécessite de fermer la boucle sur le modèle. Celui-ci peut alors être mis à jour en temps réel en fonction des données collectées par les capteurs. Toutefois, à cause des coûts élevés en calculs requis pour résoudre les modèles mécaniques physiquement réalistes, rares sont les applications qui ont réussi à les intégrer à un contrôleur en boucle fermée. Dans [Campisano 2021], les auteurs proposent un contrôleur en boucle fermée à partir du modèle Cosserat pour contrôler le bout d'un robot déformable linéaire. Dans [Zhang 2017], les auteurs contrôlent également en boucle fermée le bout d'un robot déformable, cette fois non linéaire. Pour obtenir la Jacobienne utilisée dans la commande, ils calculent d'une part la matrice de couplage entre la position de l'effecteur et les actionneurs en résolvant indépendamment, et pour chaque actionneur, les équations du modèle. D'autre

part, ils calculent la matrice de couplage entre les actionneurs. Cette dernière matrice est ensuite inversée pour obtenir la Jacobienne. Dans le cas de la manipulation d'objets déformables à deux bras, les actionneurs ne sont toutefois pas nécessairement définis selon des degrés de liberté indépendants et cette matrice n'est donc pas toujours inversible. Une autre différence avec le contrôle de robots déformables est qu'en règle général, lors de la manipulation d'objets déformables, on dispose de plus de degrés de libertés pour l'actionnement (deux bras robotiques par exemple). En supposant les contacts parfaits, on a ainsi un plus grand contrôle sur l'objet, sur lequel il y a généralement plus de points de contrôles. En contrepartie, pour une approche similaire à [Campisano 2021, Zhang 2017] le calcul de la Jacobienne est plus complexe et cette dernière aura des dimensions plus élevée. Les auteurs de [Koessler 2021] ont ainsi proposé un contrôleur en boucle fermée couplé à un modèle réduit d'éléments finis afin de contrôler la forme d'un objet linéaire. L'objet manipulé est encastré à une extrémité et saisi par un bras, réduisant ainsi le nombre de degrés de liberté lié à l'actionnement.

Afin d'éviter les coûts élevés en calcul des modèles physiquement réalistes, d'autres approches se sont concentrées sur des modèles géométriques, ou des méthodes sans modèle. En général, ces approches ont pour but d'estimer localement la matrice Jacobienne des déformations de l'objet, créant ainsi une connexion entre les actions effectuées par le robot et l'effet de ces actions sur la forme de l'objet manipulé. Dans une série de travaux [Navarro-Alarcon 2013, Navarro-Alarcon 2014, Navarro-Alarcon 2016, Navarro-Alarcon 2017], [Zhu 2018, Zhu 2021] et [Lagneau 2020], les auteurs ont appliqué des techniques d'asservissement visuel à la manipulation robotique d'objets déformables.

La matrice Jacobienne locale des déformations est initialisée, puis mise à jour à chaque itération. Celle-ci représente le lien entre une représentation bas-niveau de l'objet, que ce soit des points de contrôle [Alambeigi 2018, Hu 2018] ou un contour 2D de l'objet [Navarro-Alarcon 2017, Zhu 2021], et les commandes envoyées aux actionneurs.

La manière dont est mise à jour la matrice Jacobienne dépend ainsi des données visuelles obtenues. Les déformations de l'objet sont estimées à l'aide de caméras 2D ou 3D, puis comparées aux commandes envoyées aux actionneurs sur une même fenêtre de temps. La matrice Jacobienne à un instant t dépend donc des actions précédentes du robot. Cela pose problème car certains mouvements potentiels du robot peuvent alors rester totalement inexplorés ou encore ne plus être valables si ces derniers n'ont pas été utilisés dans la fenêtre de temps sur laquelle a été mise à jour la matrice Jacobienne.

La matrice Jacobienne représente essentiellement une linéarisation des déformations de l'objet par rapport aux mouvements du robot. Les objets

déformables ayant des comportements hautement non-linéaires, cette matrice Jacobienne doit rester locale. Ainsi, en mettant à jour uniquement les degrés de liberté excités par le robot sur la fenêtre de temps précédente, certains degrés de liberté ne sont jamais mis à jour et faussent la solution lorsqu'ils sont excités alors que l'objet est dans une configuration totalement différente. De plus, l'algorithme peut se retrouver bloqué si aucun degré de liberté excité précédemment par le robot ne permet d'offrir de solution, bien qu'elle puisse exister si l'ensemble des mouvements possibles et leur effets sur la forme de l'objet étaient pris en compte dans le calcul de solution.

Afin de remédier à ce problème, les auteurs dans [Shetab-Bushehri 2022] ont introduit un modèle géométrique, le modèle ARAP (as-rigid-as-possible) dans leur contrôleur. Au lieu de calculer la matrice Jacobienne à partir des données visuelles, ils utilisent ces dernières pour mettre à jour le modèle ARAP de l'objet, et ce dernier est à son tour utilisé pour calculer la matrice Jacobienne. Cette approche a été étendue avec la même méthode de calcul de la matrice Jacobienne et un contrôleur optimal dans [Giraud 2022, Aghajanzadeh 2022].

Toutefois, les modèles géométriques, sans réalité physique, peuvent fournir parfois des formes infaisables ou irréalistes de l'objet. Ils peuvent manquer de précision, particulièrement dans le cas de larges déformations, et ne pas capturer entièrement l'ensemble des modes de déformation de l'objet (polynôme de Legendre).

Dans ce chapitre, nous proposons une nouvelle approche en boucle fermée qui permet de contrôler la forme des objets déformables linéaires en maintenant la précision des modèles physiquement réalistes sans en avoir les inconvénients classiques. Ce contrôleur peut être utilisé indépendamment, ou de manière complémentaire avec une trajectoire générée avec la méthode présentée dans le chapitre 3. Le contrôleur est basé sur une estimation locale de la matrice Jacobienne en utilisant le modèle de poutre de Cosserat. Le contrôleur fonctionne avec les objets déformables homogènes isotropes, mais également avec des objets plus complexes comme des fils métalliques tressés et/ou avec des matériaux composites. De plus, en intégrant les équations de Cosserat à chaque itération de la boucle, on s'assure que la forme obtenue est réalisable. L'utilisation d'un modèle physique permet également d'ouvrir la possibilité d'ajouter des contraintes ou des limites sur les efforts appliqués par le robot, ou encore de modéliser les contacts extérieurs (pourvu qu'ils soient connus à l'avance).

Nous allons ensuite tester notre approche en contrôlant la forme d'objets déformables à l'aide d'un bras, puis de deux bras robotiques.

5.1 Modèle Cosserat

Dans ce chapitre, nous allons appliquer à nouveau le cadre présenté dans le chapitre 2.3. Les équations différentielles seront spécifiquement utilisées pour résoudre le problème aux valeurs initiales (IVP) :

$$\begin{aligned}
 \dot{p} &= Rv, & v &= K_t^{-1}R^T n + \tilde{v}, \\
 \dot{R} &= R\hat{u}, & u &= K_r^{-1}R^T m + \tilde{u}, \\
 \dot{n} &= -f, \\
 \dot{m} &= -\dot{p} \times n - l
 \end{aligned} \tag{5.1}$$

Ces équations 5.1 régissent l'état de l'objet $\Gamma = [p(s), R(s), n(s), m(s)]^T$. Soit la position p , l'orientation R , les forces et moments internes n et m par rapport à s le long de l'objet.

Afin d'obtenir $\Gamma = [p, R, n, m]^T$, on résout le problème aux valeurs initiales en intégrant spatialement ces équations à partir des valeurs initiales $\Gamma_0 = [p(0), R(0), n(0), m(0)]^T$ à la limite de l'objet manipulé.

5.2 Contrôle de forme

5.2.1 Formulation du problème

Pour limiter le coût en calcul, nous adoptons dans ce chapitre la formulation classique basée sur les points objectifs couplés aux points de contrôle. Cette formulation, très répandue dans la littérature [Coevoet 2017, Coevoet 2019, Shetab-Bushehri 2022, Giraud 2022], a l'avantage d'être facile à implémenter et beaucoup plus rapide à calculer que la formulation basée sur les distances développée dans le chapitre 3.

On définit deux ensembles de points couplés :

- L'ensemble des points objectifs p_{obj}^i avec $i \in [1, n_c]$, qui est une représentation bas niveau de la forme désirée de l'objet.
- L'ensemble des points de contrôle p_c^i avec $i \in [1, n_c]$, solidaires de l'objet dont on cherche à contrôler la position.

Chaque point objectif est associé à un ou plusieurs points de contrôle, et le but de l'algorithme est de minimiser l'erreur en position ε_p^i entre chaque couples de point objectif-contrôle : $\varepsilon_p = [\|p_{obj}^1 - p_c^1\|_2 \quad \dots \quad \|p_{obj}^{n_c} - p_c^{n_c}\|_2]^T = [\varepsilon_p^1 \quad \dots \quad \varepsilon_p^{n_c}]^T$.

Résoudre le problème Cosserat aux valeurs initiales est rapide et efficace si l'on connaît le type de conditions aux limites (encastré-encastré, encastré-libre ...). Il suffit d'intégrer les équations de Cosserat à partir de Γ_0 . La principale

difficulté est de trouver les conditions initiales Γ_0 qui produisent la forme désirée, autrement dit, de résoudre le problème aux valeurs limites. Dans le chapitre 3, nous avons utilisé notre algorithme d'optimisation pour trouver les conditions initiales Γ_0 permettant de produire une forme qui minimise la fonction de coût. Cette approche est trop coûteuse en calculs pour être intégrée dans un contrôleur en boucle fermée, nous devons donc trouver un autre moyen de résoudre le problème aux valeurs limites et de déterminer les conditions initiales Γ_0 désirées.

Au lieu de résoudre le coûteux problème aux valeurs limites à chaque itération, on propose de partir d'un état initial où ces limites sont connues, et de mettre à jour Γ_0 à chaque itération. A chaque fois que Γ_0 est mis à jour, l'algorithme intègre les équations de Cosserat pour obtenir une nouvelle forme de l'objet à chaque itération. Les formes successives de l'objet convergent ensuite sur un horizon de temps vers la forme désirée. Le résultat final est donc bien une solution du problème aux valeurs limites qui peut être intégrée dans une boucle fermée en temps réel.

5.2.2 Matrice Jacobienne des déformations basée sur le modèle Cosserat

La convergence de l'algorithme dépend bien sûr de la manière dont est mis à jour Γ_0 . Afin d'avoir la meilleure précision possible, on utilise à nouveau le modèle Cosserat. Plus précisément, on approxime le modèle Cosserat localement avec une matrice Jacobienne des déformations de l'objet J_d qui crée une relation entre les variations de Γ_0 et la variation de la position courante des points de contrôle p_c^i . On utilise ensuite la pseudo-inverse de J_d pour mettre à jour Γ_0 à chaque itération.

La matrice Jacobienne J_d est une approximation linéaire du modèle Cosserat pour un objet spécifique, valable localement. Calculer J_d requiert donc uniquement les conditions initiales Γ_0 de l'objet dans sa configuration actuelle. Pour chaque degré de liberté contrôlable aux points de saisie (ces derniers dépendent des conditions de saisie), on introduit une perturbation Δ_d^j , $j \in [1, m_l]$ où m_l est le nombre de degrés de liberté contrôlables. Pour un objet linéaire saisi parfaitement à ses deux extrémités, on a ainsi $m_l = 12$ degrés de liberté et la dimension de Γ_0 est nécessairement égale à 12 pour un problème de tige contrôlée-contrôlée : 3 degrés pour la position, 3 pour l'orientation et 6 pour les forces et moments internes. En intégrant les équations de Cosserat à partir des conditions initiales actuelles de l'objet, et à partir des conditions initiales perturbées, on obtient $m_l + 1$ configurations de l'objet : la configuration actuelle ainsi que m_l configurations perturbées selon chaque dimension de Γ_0 . En comparant, pour chaque point de contrôle,

la configuration perturbée de l'objet à la configuration actuelle, on peut ainsi observer indépendamment l'effet de chaque perturbation sur la position du i ème point de contrôle p_c^i avec $i \in [1, n_c]$.

On peut alors calculer par différence finie et pour chaque point de contrôle une matrice Jacobienne J_d^i des variations partielles avec $i \in [1, n_c]$ qui fournit un couplage linéaire entre chaque dimension de Γ_0 et leur effet sur la position du point de contrôle p_c^i sur l'objet. J_d est alors obtenu en regroupant chaque J_d^i pour les n points de contrôle : $J_d = [J_d^1 \ \dots \ J_d^{n_c}]^T$. L'expression suivante (équation 5.2) montre une forme générale de J_d^i pour $\Delta_d^j > 0$:

$$J_d^i = \begin{bmatrix} \frac{p_x^i(\Gamma_0 + \Delta_d^1) - p_x^i(\Gamma_0)}{\|\Delta_d^1\|} & \dots & \frac{p_x^i(\Gamma_0 + \Delta_d^{m_l}) - p_x^i(\Gamma_0)}{\|\Delta_d^{m_l}\|} \\ \frac{p_y^i(\Gamma_0 + \Delta_d^1) - p_y^i(\Gamma_0)}{\|\Delta_d^1\|} & \dots & \dots \\ \frac{p_z^i(\Gamma_0 + \Delta_d^1) - p_z^i(\Gamma_0)}{\|\Delta_d^1\|} & \dots & \frac{p_z^i(\Gamma_0 + \Delta_d^{m_l}) - p_z^i(\Gamma_0)}{\|\Delta_d^{m_l}\|} \end{bmatrix} \quad (5.2)$$

Le type de saisie du robot impacte directement le type de conditions aux limites, et donc la matrice Jacobienne. En réalité, les conditions de saisie ne sont pas parfaites, i.e. on observe un petit glissement dans une liaison encastrement, un couple de frottement dans une saisie à deux doigts, etc. Dans ces cas, la position, l'orientation, les forces et moments dans des directions spécifiques peuvent ne pas être contrôlables. De fait, la taille de m_l est inférieure à 12, et la perturbation $\Delta_d^j = 0$ pour ces degrés de liberté incontrôlables produit une variation nulle sur la position des points de contrôle. Cela résulte en une colonne de zéro dans J_d^i . En accord avec la théorie de Cosserat, si le manipulateur ne peut pas appliquer de forces ou de moments aux limites selon certaines directions, les forces et moments internes à chaque point de l'objet seront constants dans ces directions.

On peut également noter que pour une application réelle, le vecteur des valeurs initiales de l'objet Γ_0 n'est pas connu. On calcule la matrice Jacobienne à partir de l'estimation de ce vecteur qu'on appellera $\hat{\Gamma}_0$ et qu'on utilisera par la suite dans les notations.

5.2.3 Commande

Une fois que l'on a défini la matrice Jacobienne, on l'utilise pour calculer les nouvelles valeurs initiales $\Gamma_{0_{t+1}}$ qui vont être utilisées pour mettre à jour la configuration de l'objet. On utilise directement la transposée de J_d , ainsi que l'erreur ε_p à l'instant t pour calculer le nouveau vecteur des valeurs initiales $\Gamma_{0_{t+1}}$ à l'instant $t + 1$:

$$\Gamma_{0_{t+1}} = \hat{\Gamma}_{0_t} + K J_d^\dagger \varepsilon_p \quad (5.3)$$

Où $\hat{\Gamma}_{0_t}$ est l'estimation du vecteur des conditions initiales de la pose courante de l'objet, K est un paramètre qui affecte le taux de convergence vers l'objectif et la précision de l'algorithme, et J_d^\dagger représente la pseudo-inverse de J_d . K affecte directement la stabilité de l'algorithme. Par la suite, on définira notre critère de stabilité en montrant que les conditions de stabilité sont équivalentes à celles d'un problème d'asservissement visuel. Bien qu'il soit possible de montrer la stabilité locale d'une configuration, il n'est pas possible de montrer la stabilité globale en asservissement visuel [Chaumette 2006]. On définit Ω le voisinage des configurations stables autour de Γ_{0_t} . On cherche donc à déterminer $\Gamma_{0_{t+1}}$ de sorte à ce que ce dernier soit également dans Ω afin de garantir la stabilité de la méthode. Ω dépend de la complexité et de la non-linéarité de la tâche. Ainsi, plus la tâche est complexe, plus Ω est petit et plus K doit être petit.

5.2.3.1 Loi de commande robotique

Une fois les conditions initiales désirées $\Gamma_{0_{t+1}}$ calculées, on résout le problème aux valeurs initiales pour obtenir l'état mis à jour Γ_{t+1} de l'objet. On déduit alors de la position et de l'orientation de la forme désirée de l'objet aux points de saisie, la position et l'orientation désirées des effecteurs du robot. On formule alors notre loi de commande proportionnelle pour calculer les vitesses linéaires et angulaires du ou des manipulateurs permettant de le ou les amener vers leur position et orientation désirées, (voir équation 5.4) :

$$\begin{aligned} v_t &= -K_t e_t, \\ v_a &= -K_a e_a \end{aligned} \tag{5.4}$$

Où v_t et v_a sont les vitesses linéaires et angulaires du manipulateur, K_t et K_a les matrices de gain associées, diagonales et positives, et e_t et e_a les erreurs en position et orientation du ou des manipulateurs.

5.2.4 Perception

Pour calculer la matrice Jacobienne, on a besoin de connaître l'état de l'objet à l'instant courant, et plus précisément, les conditions initiales $\Gamma_0 = [p_0, R_0, n_0, m_0]^T$. Ces conditions initiales correspondent à la forme courante de l'objet et sont cruciales pour calculer la Jacobienne J_d car elles constituent le point de référence par rapport auquel sont calculées les configurations perturbées (voir équation 5.2). Il est donc important d'avoir une estimation précise de ces conditions initiales qui puisse être intégrée à une boucle de contrôle en temps réel. Bien que $p(0)$ et $R(0)$ puissent être facilement déduits de la position et de l'orientation du robot au point de

saisie, le principal défi ici est d'estimer les efforts et moments internes au point initial $n(0)$ et $m(0)$. On propose deux méthodes avec chacune ses avantages et inconvénients.

5.2.4.1 Capteurs de force

La première méthode est la plus simple, en tout cas en théorie. Il suffit d'équiper l'un des bras robotiques avec un capteur de force et de mesurer directement les forces et moments à l'une des limites pour obtenir $n(0)$ et $m(0)$. Le problème est que n et m , contrairement à p et R , sont extrêmement sensibles aux bruits de mesures. De légères erreurs sur les mesures peuvent générer des écarts importants sur la forme obtenue de l'objet par rapport à la forme réelle. Il est donc nécessaire de filtrer le signal obtenu, ce qui ajoute un délai dans la boucle.

5.2.4.2 Vision

Cette approche requiert davantage d'étapes, mais est plus robuste au bruit. Elle consiste à utiliser la vision pour mesurer la position des points de contrôle p_c^i . Pour estimer le vecteur des conditions initiales de la pose courante $\hat{\Gamma}_0$ à partir des mesures de p_c^i , on utilise les conditions initiales et la matrice Jacobienne calculées à l'itération précédente (voir équation 5.5) :

$$\hat{\Gamma}_{0_t} = \hat{\Gamma}_{0_{t-1}} + K J_{d_{t-1}}^\dagger \varepsilon_p \quad (5.5)$$

Cette approche est plus facile à mettre en place car les mesures de p et R sont beaucoup moins sensibles au bruit. Toutefois, à l'instant courant, on estime les conditions initiales Γ_0 à partir de l'estimation des conditions initiales à l'étape précédente, qui sont elles-mêmes calculées à partir de l'estimation des conditions initiales à l'étape précédente et ainsi de suite. Il est donc nécessaire d'avoir une estimation précise de Γ_0 à l'étape initiale de l'algorithme. Il n'est pas nécessaire d'avoir une précision absolue, il y a toujours une légère erreur sur l'estimation de Γ_0 . Tant que l'erreur ne change pas la dynamique des déformations entre l'objet réel et la forme estimée, l'algorithme va resynchroniser de lui même Γ_0 avec la valeur réelle. Plus précisément, les $\Gamma_{0_{t+1}}$ calculés si $\hat{\Gamma}_{0_t}$ a une erreur, vont être très proches les uns des autres et l'objet va ainsi ralentir et "attendre" que Γ_0 "rattrape" la valeur réelle.

En fonction des algorithmes de vision utilisés, cette approche peut également être sensible aux occultations et être plus lourde en calculs que l'approche basée sur les capteurs de force. C'est cette approche que nous utiliserons par la suite.

5.2.5 Contrôle en boucle fermée

On représente la boucle de contrôle sur la figure 5.1 et sur l'algorithme 1. Une itération de la boucle peut être résumée à :

- A partir de la méthode de perception utilisée, définir l'erreur ε_p et estimer les conditions initiales de l'objet dans sa configuration actuelle $\hat{\Gamma}_0$ (voir équation 5.5).
- Utiliser $\hat{\Gamma}_0$ et les équations du modèle Cosserat pour calculer la matrice Jacobienne des déformations de l'objet pour la pose courante.
- Utiliser la pseudoinverse de la matrice Jacobienne et résoudre le problème pour mettre à jour les conditions initiales $\Gamma_{0_{t+1}}$.
- Résoudre le problème aux valeurs initiales en intégrant les équations de Cosserat à partir de $\Gamma_{0_{t+1}}$ pour obtenir l'état Γ_{t+1} mis à jour de l'objet.
- A partir de Γ_{t+1} , déterminer les commandes à envoyer au robot pour mettre à jour l'objet réel.
- Envoyer les commandes au robot.
- Répéter les étapes précédentes jusqu'à la convergence de l'objet vers la forme souhaitée.

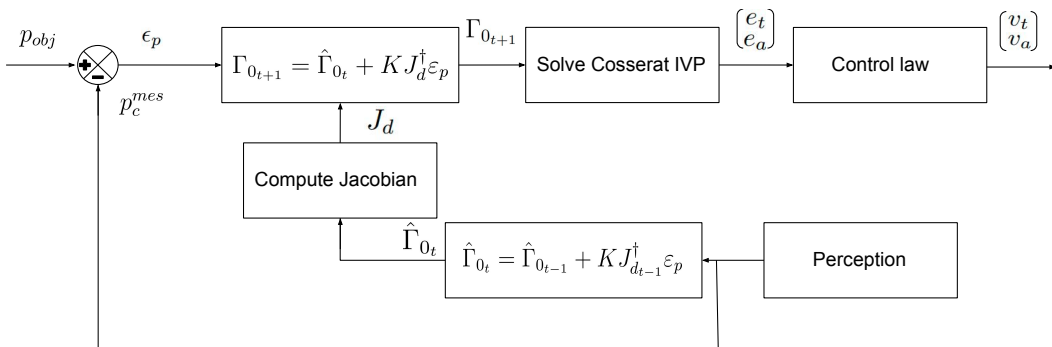


FIGURE 5.1 – Schéma bloc d'une boucle de contrôle.

5.2.6 Problème bien posé et critère de stabilité

Le problème direct de Cosserat est bien posé [Antman 2005]. Pour chaque ensemble de conditions initiales Γ_0 , il existe un unique état Γ obtenu en intégrant les équations de Cosserat, et la solution Γ varie continuellement si Γ_0 varie continuellement. La seconde partie de cette affirmation est importante pour nous car elle signifie qu'en mettant à jour les conditions initiales de manière continue au cours de la manipulation, alors la forme obtenue de l'objet va également être mise à jour de manière continue. Les conditions de stabilité peuvent alors être réduites aux conditions de stabilité classiques du contrôle visuel posées dans [Chaumette 2006], c'est-à-dire que la loi de commande est

Algorithm 1 Control loop at step t

```

 $J_{d_{t-1}}$  // Jacobian computed at step t-1
 $n_{0_{t-1}}$  // internal forces computed at step t-1
 $m_{0_{t-1}}$  // internal moments computed at step t-1

while l=1 do
   $p_c^{mes} = \text{getfromvision}()$ 
   $\varepsilon_p = p_{obj} - p_c^{mes}$ 
   $\varepsilon_p^i = \max(\varepsilon_p)$  // select subproblem

   $\hat{\Gamma}_{0_t} = \hat{\Gamma}_{0_{t-1}} + K J_{d_{t-1}}^\dagger \varepsilon_p$  // estimate current initial conditions

   $J_{d_t} = \text{ComputeJacobian}(\Gamma_{0_t})$ 
   $\Gamma_{0_{t+1}} = \hat{\Gamma}_{0_t} + K J_{d_t}^\dagger \varepsilon_p$ 
   $[e_t, e_a]^T = \text{SolveCosserratIVP}(\Gamma_{0_{t+1}})$  // EE position and orientation error
   $[v_t, v_a] = \text{ControlLaw}(e_t, e_a)$  // EE velocities
end while

```

au mieux stable localement autour du point d'équilibre. Cela rejoint ce que l'on a décrit précédemment sur le domaine de validité de la matrice Jacobienne Ω . La vitesse des manipulateurs doit également être "assez basse" afin de respecter au mieux l'hypothèse quasi-statique en déformant l'objet d'un point d'équilibre vers le suivant.

On définit ainsi notre critère de stabilité de manière similaire à ce qu'ont proposé les auteurs dans [Gilbert 2016, Till 2017a, Edelmann 2017] plus tard repris dans [Campisano 2021] et qui rejoint les conditions de stabilité locale posées dans [Chaumette 2006]. De la Jacobienne J_d en $s = 0$, on extrait la partie liée à la variation des efforts et moments internes et on définit la matrice de rigidité effective :

$$M = J_d^\dagger(:, 6 : 12) \quad (5.6)$$

Contrairement aux travaux précédents sur les robots déformables, on définit la rigidité effective M à la base de l'objet et non à l'extrémité car c'est en ce point que l'on définit Γ_0 . De plus, on n'utilise pas la pseudoinverse de J_d directement car J_d contient à la fois des informations sur la variation de la position, de l'orientation et des efforts appliqués en $s = 0$.

On décompose ensuite M en valeurs singulières de sorte à trouver ses valeurs propres :

$$M = V\Sigma^\dagger U^T \quad (5.7)$$

Où

$$\Sigma^\dagger = \text{diag}\left(\frac{1}{\sigma_1}, \dots, \frac{1}{\sigma_{m_l}}\right) \quad (5.8)$$

$\sigma_1, \dots, \sigma_{m_l}$ sont les valeurs propres de $J_d(:, 6 : 12)$ et doivent rester strictement positives au cours de la manipulation de sorte à éviter les configurations instables. Cela indique que les forces appliquées par le préhenseur pour une configuration d'équilibre vont à l'encontre des forces de rappel élastiques appliquées par l'objet en $s = 0$, maintenant ainsi un équilibre stable [Campisano 2021]. On définit ainsi le voisinage stable $\Omega : \{\Gamma_{0_{t+1}} \in \mathbb{R}^{12} \mid \min([\sigma_1, \dots, \sigma_{m_l}]) > 0\}$.

De plus, si la matrice Jacobienne J_d est de rang plein, alors le système (ensemble des positions des points de contrôle sur l'objet) est commandable.

En théorie, le problème inverse est également bien posé (en tout cas de la manière dont il est formulé dans [Rucker 2010]). Mais en réalité il est loin de l'être et bien que Γ varie toujours continuellement avec Γ_0 , on ne peut pas garantir l'unicité de la solution. La principale raison est que dans la formulation du problème, la viscosité est totalement négligée. C'est une hypothèse très forte qui ne se vérifie pas en réalité. Pour s'en apercevoir, il suffit de perturber un objet encastré-encastré, légèrement visqueux (caoutchouc par exemple) en équilibre. Il est tout à fait possible que l'objet ne revienne pas parfaitement à sa condition d'équilibre précédente, mais à une nouvelle condition d'équilibre proche. Cela implique que (i) La solution n'est pas unique en réalité, (ii) La solution dépend du chemin pris par l'objet. Dans l'écrasante majorité des cas, les solutions multiples sont proches les unes des autres, et cela ne pose pas de problème sur la stabilité. Au contraire, la viscosité introduit un comportement d'amortissement qui a des propriétés stabilisantes.

Toutefois, il existe toujours certaines configurations chaotiques bien connues, notamment les configurations avec du flambement [O'Reilly 2011] (voir figure 3.9) autour desquelles il faut toujours faire attention.

5.3 Validation expérimentale

5.3.1 Dispositif expérimental

Afin de valider notre approche, on utilise deux bras Franka Emika pour manipuler différents objets linéaires. On a testé l'algorithme pour deux types de conditions aux limites et trois objets différents avec différentes propriétés :



FIGURE 5.2 – Objets utilisés dans les expériences

- Objet 1 : Le même objet que dans le chapitre 3, une bande de caoutchouc de section carrée. Le matériau est homogène et isotrope et l'objet est très flexible.
- Objet 2 : Un câble d'acier tressé de section circulaire, très rigide. Le matériau est non homogène et non isotrope.
- Objet 3 : Un câble d'acier tressé gainé de section circulaire avec un diamètre inférieur à celui de l'objet 2, le rendant moins rigide. Le matériau est composite, non homogène et non isotrope.

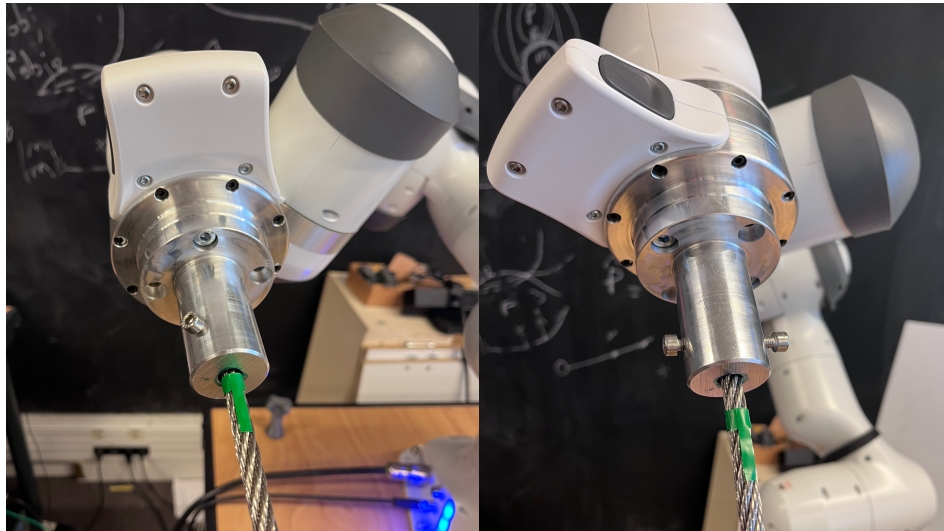
La photo 5.2 représente les trois objets linéaires utilisés.

Pour le retour visuel, on utilise un ensemble de caméras Optitracks calibrées, qui suivent la position d'un ensemble de marqueurs placés sur les points de contrôle de l'objet. N'importe quelle méthode visuelle peut être utilisée ici du moment qu'elle fournit en temps réel la position des points de contrôle, l'approche est modulaire à ce niveau. On a décidé d'utiliser un système Optitrack à la fois pour sa précision (<1 mm), et pour sa robustesse aux occultations. L'inconvénient est le temps de calcul nécessaire plus élevé que pour d'autres approches. Au vu de la rapidité de calcul de notre algorithme (plus de 100 Hz par boucle), on peut se permettre d'avoir un temps alloué à la vision plus élevé afin d'avoir de meilleures performances.

On utilise un ordinateur avec une unité CPU Intel Xeon Silver 4214R et le chipset graphique standard. On utilise ROS melodic pour communiquer avec le robot et l'algorithme tourne confortablement à une fréquence de 10 Hz, où l'écrasante majorité de la charge de calcul vient de la vision.

On a également conçu deux effecteurs fixés aux robots permettant de respecter au mieux les conditions d'encastrement aux limites, c'est-à-dire un déplacement relatif nul entre le point de saisie sur l'objet et celui sur le robot. Bien que ces fixations permettent de respecter ces hypothèses pour les objets 1 et 3, on constate au cours des expériences que ce n'est pas le cas pour l'objet 2. On utilise un système de vis qui maintient l'objet en place, les objets 1 et 3 ont une surface molle qui permet à la vis de s'enfoncer et de fixer l'objet, mais la surface de l'objet 2 est uniquement métallique et on a constaté des glissements

pendant la manipulation. Pour les expériences réalisées, ces glissements étaient de l'ordre de 1 à 2 *cm* le long de l'axe de l'objet, et jusqu'à $\pi/2$ autour de l'axe. Ces glissements sont mis en évidence sur la figure 5.3 où on observe la position relative de l'objet par rapport à la fixation, avant et après la manipulation.



Fixation de l'objet avant la manipulation - Après la manipulation - mise en évidence des glissements relatifs entre l'objet et le point de fixation

FIGURE 5.3

5.3.2 Protocole et configuration initiale de l'objet

On définit le protocole suivant (voir figure 5.4) : on détermine l'ensemble des points objectifs p_{obj}^i qui représentent la forme désirée de l'objet. Pour s'assurer de la faisabilité de cette forme et éviter les problèmes liés aux couplages relatifs entre différents points de l'objet (détaillés dans le chapitre 3), on amène manuellement l'objet vers la forme souhaitée à l'aide du mode de guidage des robots collaboratifs Franka Emika. Lorsque l'objet est dans la configuration souhaitée, on enregistre la position des points de contrôle et on utilise ces positions comme p_{obj}^i . L'ensemble des formes désirées impliquent des déformations en trois dimensions. On amène ensuite, toujours à la main, l'objet dans sa configuration initiale et on lance le contrôleur. On enregistre l'erreur ε_p en fonction du temps jusqu'à la convergence.

La configuration initiale du manipulateur définit la configuration initiale de l'objet. Comme spécifié dans la section Perception, l'algorithme peut commencer de n'importe quelle configuration initiale du moment que le vecteur Γ_0 associé à cette configuration est connu précisément.

La configuration initiale de l'objet n'est jamais pré-définie à l'avance. Avant chaque test, on guide l'objet pour l'emmener dans une configuration quelconque qu'on va considérer comme point de départ. On estime ensuite les valeurs initiales Γ_0 en utilisant un algorithme basé sur la méthode de tir similaire à celui présenté dans le chapitre 3. On mesure la position des points de contrôle p_{obj}^i , et en résolvant le problème aux valeurs limites, on trouve les valeurs initiales Γ_0 qui correspondent à la configuration initiale de l'objet. Une fois ces valeurs trouvées, on peut ensuite lancer l'algorithme et les mettre à jour. Alternativement, on pourrait définir une configuration de l'objet et partir de cette dernière précisément à chaque essai, mais cela implique de toujours commencer à la même configuration.

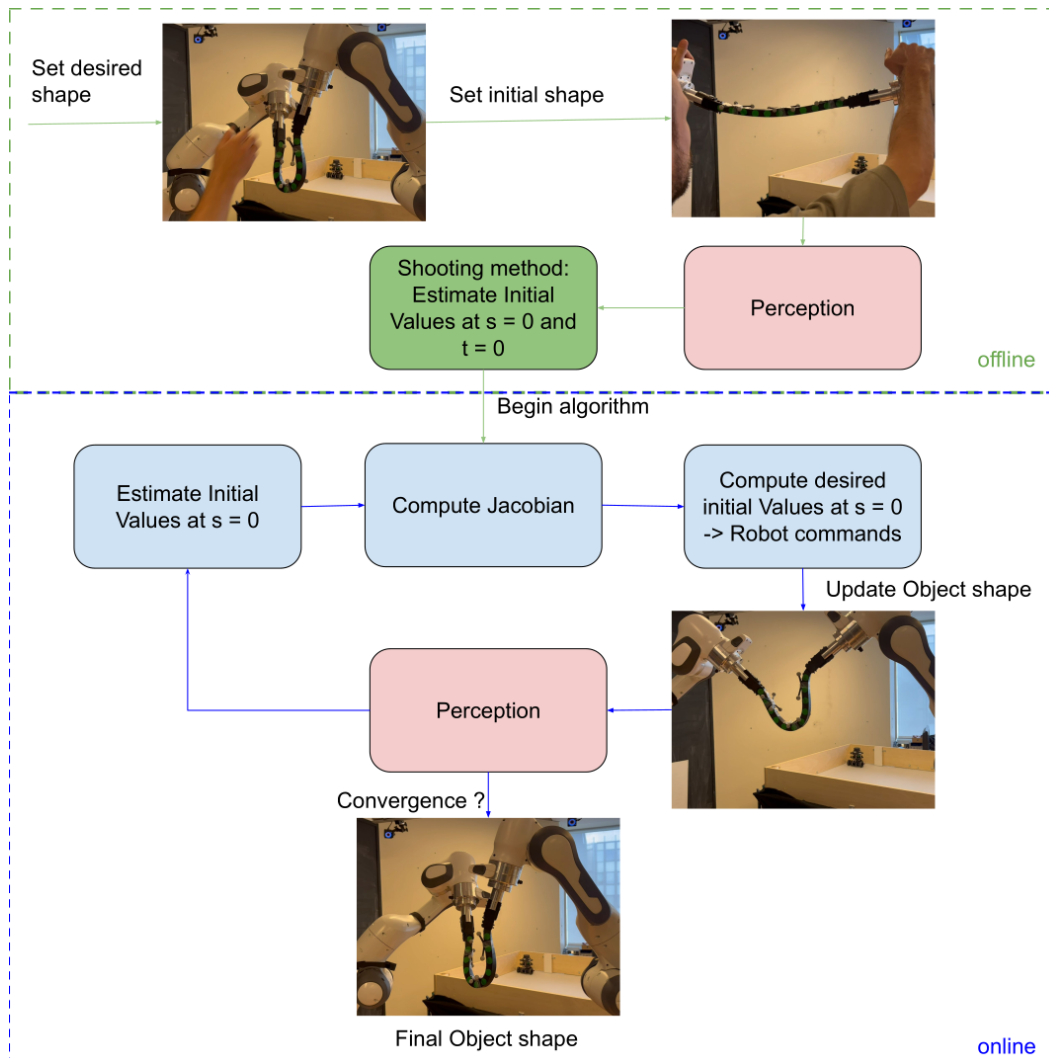


FIGURE 5.4 – Protocole expérimental

Le résultat est que les configurations initiales diffèrent légèrement à

chaque essai. Bien que la solution obtenue avec la méthode de tir soit très précise sur la position et l'orientation, elle est beaucoup plus sensible aux erreurs pour l'estimation des forces et moments internes. Plus la solution est complexe (beaucoup de modes de déformation), plus l'objet sera rigide ce qui implique une configuration instable, et moins les efforts et moments internes initiaux $n(0)$ et $m(0)$ seront précis. La quasi totalité des échecs viennent d'une mauvaise estimation des valeurs initiales à l'initialisation. On essaie donc de ne pas définir de conditions initiales trop difficiles à estimer lors de la résolution du modèle inverse.

5.3.3 Manipulation à un bras

Dans un premier temps, on teste l'algorithme en manipulant un objet déformable avec un seul bras Franka Emika. On manipule le même objet que dans le chapitre 3 (objet 1 sur la figure 5.2). Les conditions aux limites sont contrôlé-encasté, avec une contrainte de position et d'orientation constantes dans le repère global sur les valeurs initiales.

Le dispositif utilisé est similaire à celui du chapitre 3. L'objet est saisi par le robot Franka à l'une des limites, et encasté à l'aide d'un serre-joint à l'autre. La position et l'orientation de l'objet dans Γ_0 sont fixées, on a donc six colonnes de zéros dans la matrice Jacobienne. Calculer la pseudoinverse de matrices creuses peut être très compliqué. On limite alors la dimension de la matrice à la dimension du problème en supprimant les colonnes nulles. On se retrouve ainsi avec une matrice Jacobienne J_d non creuse de $3n_c$ lignes correspondant à la position des n points de contrôle, et six colonnes, correspondant à la dimension de $n(0)$ et $m(0)$. On applique le protocole défini dans la section précédente et on affiche un échantillon des résultats obtenus sur la figure 5.5. On affiche les configurations initiales, finales et désirées de l'objet, ainsi que l'erreur ε_p en fonction du temps.

Sur la figure 5.6 on observe, à chaque itération, la superposition entre les données visuelles et les formes calculées par l'algorithme au cours de la manipulation. Sur la sous-figure 5.6a, on affiche la forme initiale estimée de l'objet en rouge, ainsi que les points de contrôles p_c^i (ronds rouges o) dont la position est suivie et donnée par les caméras optitracks, ainsi que les points objectifs p_{obj}^i (étoiles rouges *). A chaque itération de la boucle, on récupère p_c^i (ronds rouges o), qu'on utilise pour estimer les conditions initiales Γ_0 qui correspondent à la forme mesurée de l'objet. On calcule ensuite la matrice Jacobienne J_d qu'on utilise pour définir Γ_0^d . On intègre ensuite les équations de Cosserat à partir de Γ_0^d pour obtenir Γ^d , qu'on trace en bleu sur la figure. On peut ainsi observer et comparer à chaque itération la forme de l'objet calculée

par l'algorithme, à celle obtenue par les caméras. On observe par exemple une légère désynchronisation entre les données visuelles et la forme calculée sur la sous-figure 5.6b, qui est rattrapée sur la sous-figure 5.6d. Ce phénomène est celui mentionné et expliqué dans la partie Perception. On affiche également en dessous des graphiques l'erreur ε_p ainsi que les erreurs ε_p^i individuelles entre chaque couple points de contrôle-objectifs.

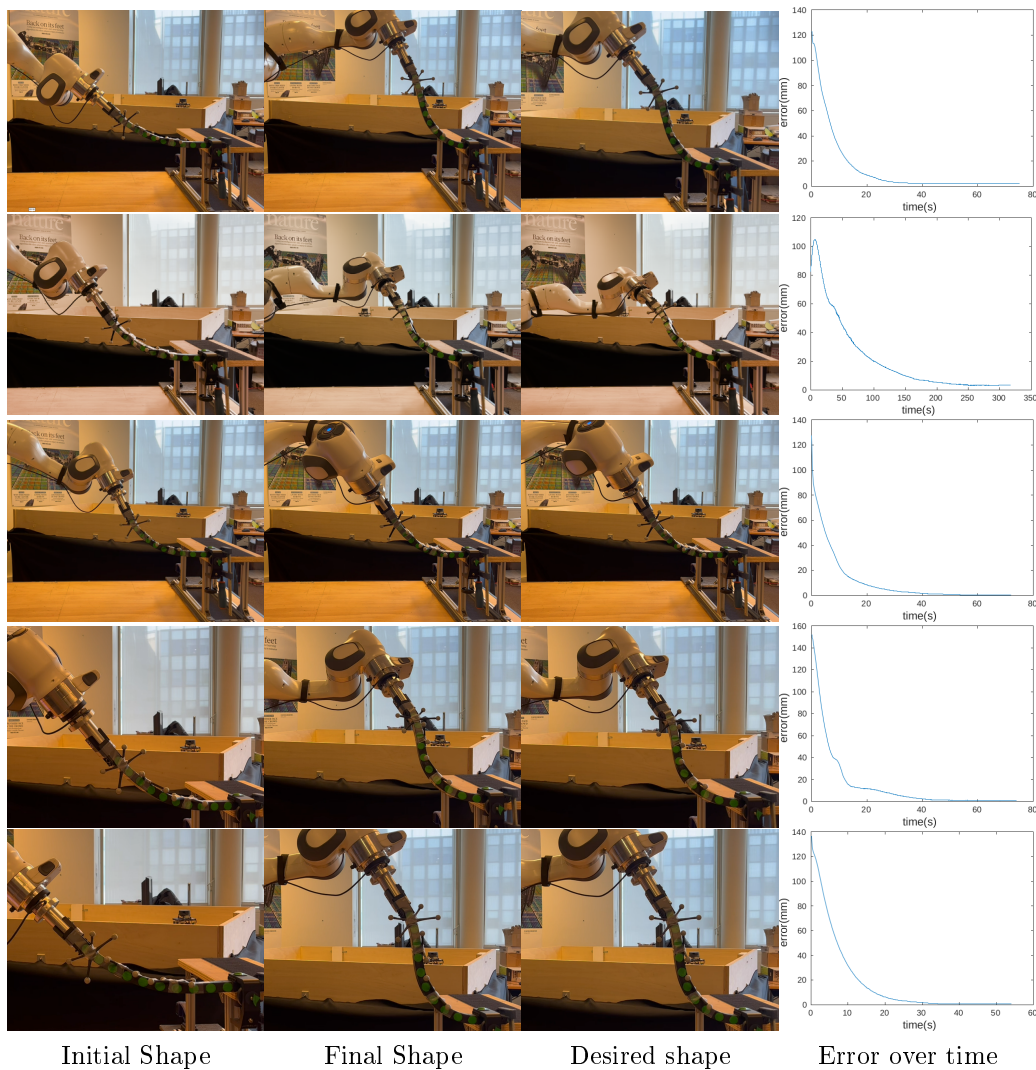


FIGURE 5.5 – Contrôle en boucle fermée de la forme de la bande élastique à l'aide d'un bras robotique

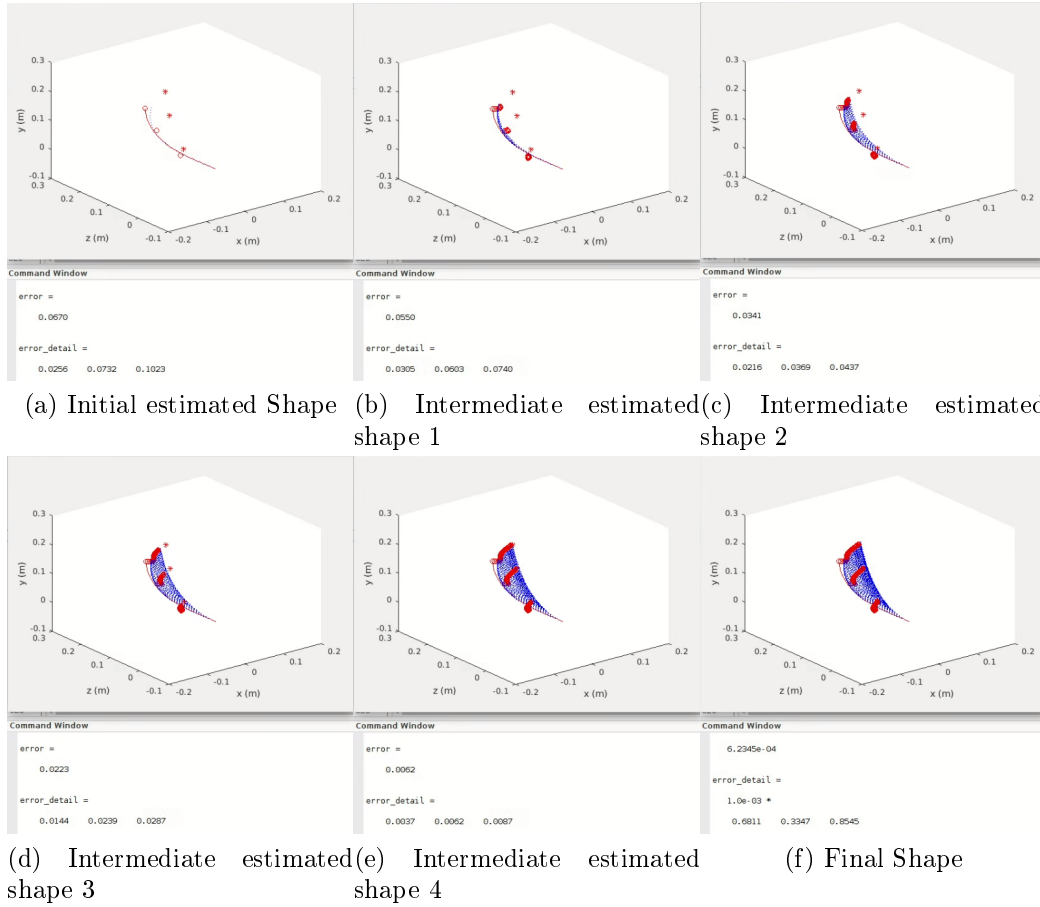


FIGURE 5.6 – Evolution de la forme calculée de l'objet au cours du temps

- : Estimation initiale de la forme de l'objet

- - : Forme calculée de l'objet après mise à jour des conditions initiales

o : Points de contrôle p_c^i , donnés par les caméras* : Points objectifs p_{obj}^i

5.3.4 Manipulation à deux bras

5.3.4.1 Contraintes sur les conditions initiales levées

La manipulation à un bras est intéressante comme première étape de validation, mais elle est limitante en termes de formes possibles. Pour réaliser les applications les plus complexes, on a besoin d'ajouter un deuxième bras pour saisir l'objet à ses deux limites. De plus, ajouter des contraintes de déplacement nul et d'orientation constante sur les configurations initiales réduit de moitié la dimension du problème, le rendant plus simple à résoudre. On ajoute donc un second bras Franka Emika de sorte à saisir l'objet à ses deux limites. Les conditions aux limites pour l'objet sont encastré-encastré et

on n'ajoute pas de contraintes sur les conditions initiales. La dimension de J_d est donc de $3n_c$ lignes et 12 colonnes pour les conditions initiales sur p , R , n et m .

5.3.5 Résultats pour différents objets complexes

On a testé notre protocole sur les trois objets présents sur la figure 5.2. Une vue détaillée de la manipulation de chaque objet est montrée sur la figure 5.7. De plus, une comparaison entre les formes finales et initiales ainsi que ε_p en fonction du temps est représentée sur la figure 5.8. Pour chaque test réalisé, l'objet converge vers la forme désirée avec une erreur de moins de 1mm, ou proche de 1mm. Les courbes de l'erreur ont un profil exponentiel, la plus grosse partie de la déformation se produit dans les premières secondes de la manipulation.

Les paramètres élastiques des objets 1 et 3 (figure 5.2) ont été estimés avec la méthode présentée dans le chapitre 3. Pour la bande élastique, ces paramètres semblent cohérents. Toutefois pour le câble tressé avec une gaine, l'algorithme nous renvoie des valeurs très différentes à chaque essai. Ce n'est pas surprenant si l'on considère que ces constantes ont un sens physique principalement pour les matériaux homogènes isotropes. On a tout de même fait la moyenne des valeurs obtenues en supposant que l'erreur est importante sur ces paramètres. Pour l'objet 2, on a utilisé exactement les mêmes paramètres que pour l'objet 3 bien que les deux objets aient un comportement et une géométrie différentes (l'objet 2 est beaucoup plus rigide que l'objet 3, si l'on prend en compte la gaine, les deux ont le même diamètre). Les paramètres utilisés sont représentés sur le tableau 5.1.

On utilise ces approximations grossières car :

- On n'a pas de moyen disponible pour estimer précisément les paramètres élastiques équivalents de matériaux composites non homogènes et non isotropes. De plus, de nombreuses hypothèses sur les symétries de l'objet ne sont plus satisfaites, il serait donc nécessaire d'estimer ces paramètres dans toutes les directions de déformation possibles, ce qui est coûteux en temps et en ressources.
- Cela nous permet de prouver que notre méthode est robuste à l'estimation des paramètres élastiques de l'objet.

On remarque que même avec une estimation de ces paramètres complètement fautive, et avec du glissement aux limites pour l'objet 2, l'erreur converge tout de même sous le millimètre. La précision des modèles physiquement réalistes dépend fortement de la précision de l'estimation de ces paramètres. Or, c'est principalement l'amplitude des déformations qui est affectée, la direction des déformations varie beaucoup moins. Lorsqu'on calcule

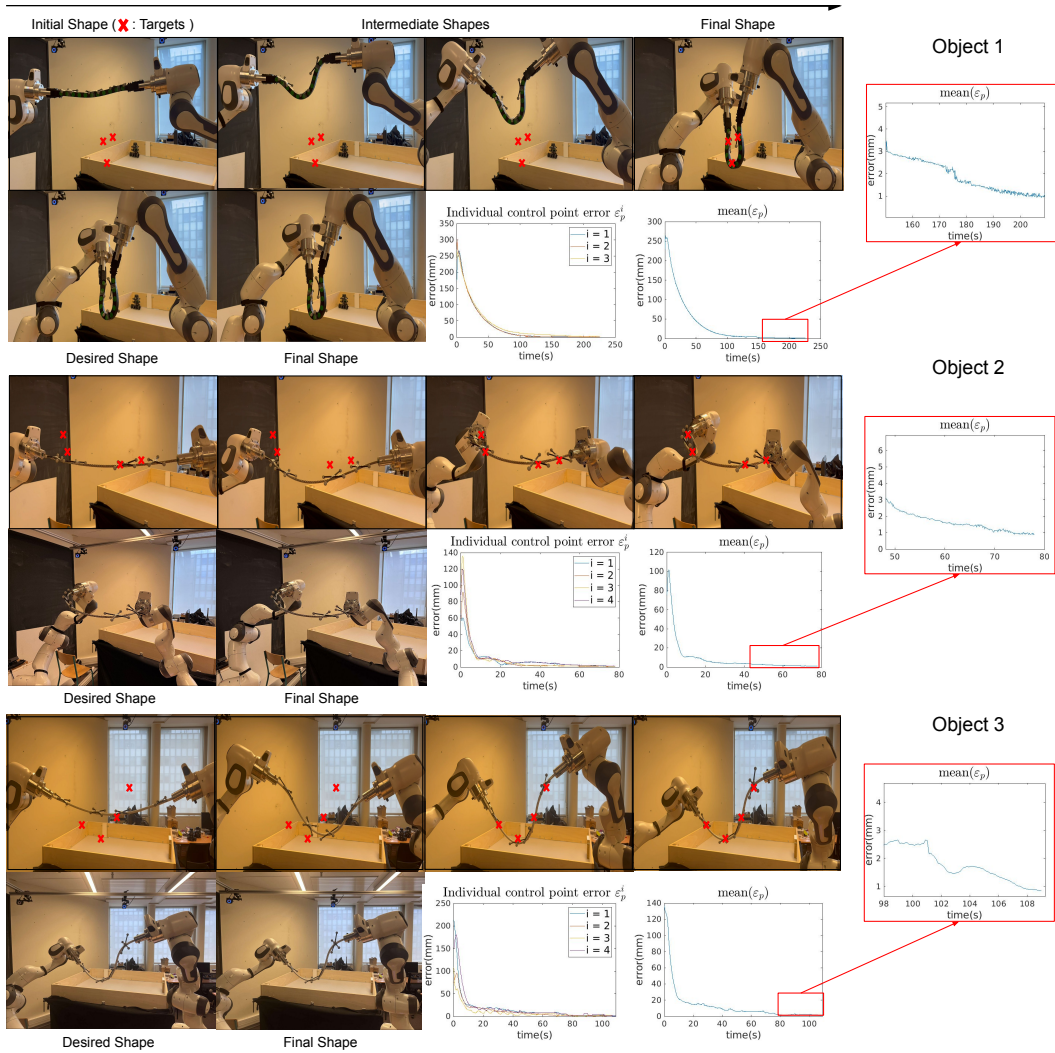


FIGURE 5.7 – Contrôle en boucle fermée de la forme des trois objets à l'aide de deux bras robotiques

J_d par différences finies entre différentes configurations statiques de l'objet, c'est exactement ce type de comportement que l'on essaie de capturer. Ce phénomène est illustré sur la figure 5.9. On y observe l'impact des paramètres élastiques sur la forme d'un objet linéaire. Ces formes sont obtenues en intégrant les équations de Cosserat à partir d'un vecteur Γ_0 constant, en faisant varier uniquement le module d'Young E et le coefficient de poisson ν . Bien que l'amplitude des déformations diffère logiquement, les modes de déformation excités sont les mêmes peu importe les paramètres élastiques. De plus, si on suit l'évolution d'un point précis entre chacune des configurations affichées, la direction de ce point ne varie quasiment pas.

Cela explique pourquoi en se concentrant sur la direction dans laquelle

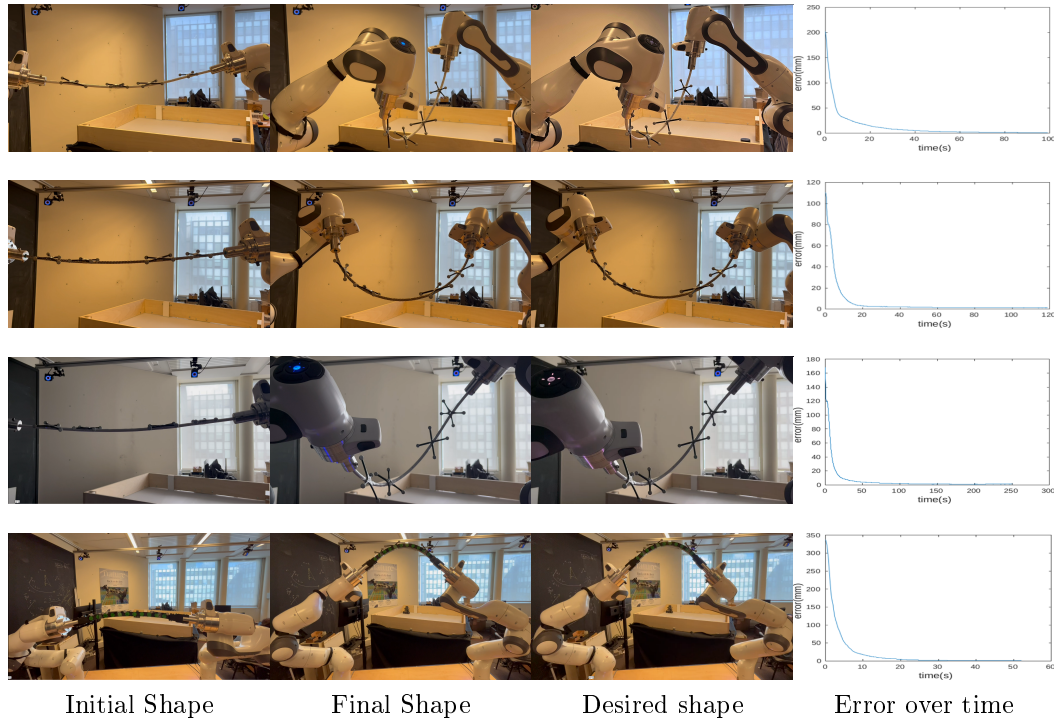


FIGURE 5.8 – Autres exemples de contrôle en boucle fermée de la forme des trois objets à l'aide de deux bras robotiques

vont évoluer les points de contrôle, et non sur leur position précise, on obtient une méthode robuste aux erreurs d'estimation de nombreux paramètres. A chaque itération, le contrôleur calcule donc la direction que les points de contrôle doivent emprunter pour atteindre les points objectifs, et s'en rapproche itérativement jusqu'à les atteindre. On n'a ainsi pas observé d'écart de performance du contrôleur en manipulant l'objet homogène isotrope et en manipulant les câbles non isotropes et non homogènes.

En comptant les tests présentés au dessus, on a conduit un total de 35 tests. L'ensemble des erreurs finales après convergence sont affichées sur la figure 5.10. L'ensemble des tests impliquent des déformations en trois dimensions, avec différents nombres et emplacements des points de contrôle, ainsi que différentes configurations de l'objet à l'initialisation et différentes formes désirées. Les tests ont tous été conduits en suivant le protocole présenté précédemment. Sur les 35 tests, 6 seulement convergent avec une erreur finale supérieure à 1 mm, et on a également eu 4 cas d'échec. On a considéré qu'un test est échoué si la convergence finale est supérieure à 3mm. Les causes d'échec sont :

- Une mauvaise estimation de Γ_0 à l'initialisation.

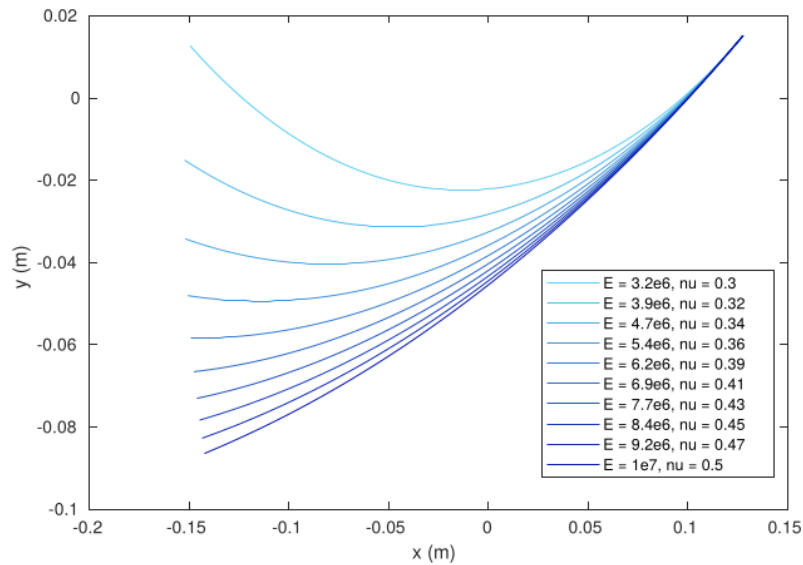


FIGURE 5.9 – Evolution de la forme d'un objet en faisant varier uniquement les paramètres élastiques

- Des objectifs inatteignables. Cela peut arriver, notamment avec les objets les plus rigides, si le robot n'arrive pas à déployer assez d'efforts pour amener l'objet vers la forme objectif. Bien qu'on amène nous-même au préalable l'objet dans cette forme, on peut déployer bien plus d'efforts lorsque le robot est en mode guidage que lorsqu'il bouge de lui-même.

5.3.6 Configurations instables et temps de convergence

Le temps de convergence dépend principalement de la complexité de la forme désirée. Généralement, plus l'énergie élastique est élevée dans l'objet, moins la configuration est stable. Les objets rigides sont donc généralement moins stables que les objets souples pour une déformation identique. Cela les rend plus difficiles à manipuler précisément car ils se déforment plus souvent dans des configurations d'équilibre instable. En contrepartie, cette rigidité rend les modes de déformation complexes plus difficiles à exciter, ils ont donc tendance à être plus faciles à modéliser.

Lorsqu'on essaie d'atteindre des formes complexes qui impliquent une énergie élastique élevée dans l'objet, le voisinage Ω est beaucoup plus petit. On doit donc faire de plus petits pas à chaque itération de sorte à toujours rester dans ce voisinage. Le paramètre de contrôle K défini dans l'équation 5.3 sera donc élevé et l'algorithme devra faire plus d'itérations pour arriver à une même configuration, augmentant donc automatiquement le temps de convergence.

Parameter	Object 1	Cables
E	$3.2e^6 \text{ N/m}^2$	$1e^8 \text{ N/m}^2$
ν	0.5	0.3
$G = \frac{E}{2(1+\nu)}$	$1.06e^6 \text{ N/m}^2$	$3.84e^7 \text{ N/m}^2$
EI_x	0.04 N.m^2	0.032 N.m^2
EI_y	0.04 N.m^2	0.032 N.m^2
GJ	0.028 N.m^2	0.0248 N.m^2
ρ	1392 kg/m^3	2850 kg/m^3
L	0.29 m	0.625 m
h	0.02 m	...
b	0.02 m	...
r	...	0.0045 m
g	9.81 m/s^2	9.81 m/s^2

TABLE 5.1 – Paramètres des objets manipulés

L'algorithme peut également trouver des configurations d'équilibre stable statiquement, mais instable dynamiquement. Il est donc judicieux dans ces cas de réduire les gains sur la vitesse du robot pour respecter au mieux l'hypothèse quasi-statique.

La rigidité effective d'un objet dépend des propriétés des matériaux qui le composent, de sa géométrie et de sa configuration (voir équation 5.7). L'objet 2 sur la figure 5.2 est de loin le plus rigide, le robot arrive à peine à le déformer. L'objet 1 est le moins rigide et l'objet 3 est entre les deux. Une configuration d'un objet peut être stable dans une direction mais complètement instable dans une autre. La tension intrinsèque dans les objets plus rigides les empêche de se déformer sous leur propre poids vers des configurations stables.

Pour ce qui est des configurations les plus instables, voir chaotiques comme le flambement [O'Reilly 2011] (voir figure 3.9), l'algorithme va toujours éviter ces configurations à moins qu'on lui rajoute des contraintes sur le calcul de la solution. Se retrouver dans ces configurations implique d'appliquer des actions très spécifiques sur l'objet, par exemple une compression pure dans le cas du flambement. Lorsqu'on calcule la matrice Jacobienne, on prend en compte l'ensemble des actions possibles qui ont un effet sur la position des points de contrôle. Il est donc impossible, sans rajouter de contraintes sur la matrice Jacobienne, de trouver une telle solution si spécifique.

5.3.7 Répétabilité

Pour finir, on teste également la répétabilité de l'approche. On essaie d'atteindre la même configuration pour cinq essais successifs. La configuration

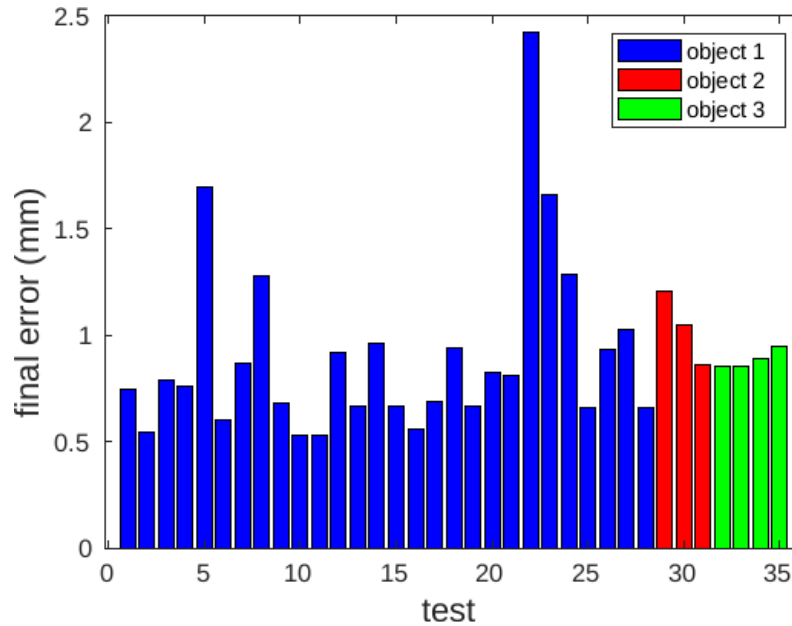


FIGURE 5.10

de l'objet à l'initialisation est légèrement différente à chaque fois, et l'erreur à l'initialisation varie entre 90 et 160mm sur les cinq essais. Les formes finales, ainsi que l'erreur en fonction du temps sont représentées sur la figure 5.11. Le profil de l'erreur, ainsi que les formes obtenues sont très similaires pour les cinq essais successifs. A l'exception de l'essai 4 où l'erreur finale est de 2 mm, l'ensemble des autres essais converge vers la forme désirée avec une erreur inférieure à 1 mm avec un profil identique peu importe la configuration initiale.

5.4 Conclusion et perspectives

Dans ce chapitre, nous avons présenté notre approche permettant de contrôler la forme d'un objet déformable en boucle fermée et en trois dimensions. A chaque itération, on estime les valeurs initiales de l'objet à partir des données visuelles, on calcule ensuite la matrice Jacobienne des déformations à partir de ces valeurs initiales et du modèle Cosserat. On utilise la pseudoinverse de cette matrice Jacobienne pour mettre à jour les valeurs initiales qu'on intègre pour obtenir la forme mise à jour de l'objet. On peut alors en déduire les commandes à envoyer aux robots. On a expérimentalement mis en évidence la précision de l'approche en atteignant des formes complexes avec une erreur sous le millimètre et une convergence exponentielle, même en manipulant des objets non isotropes et non homogènes dont on ne connaît

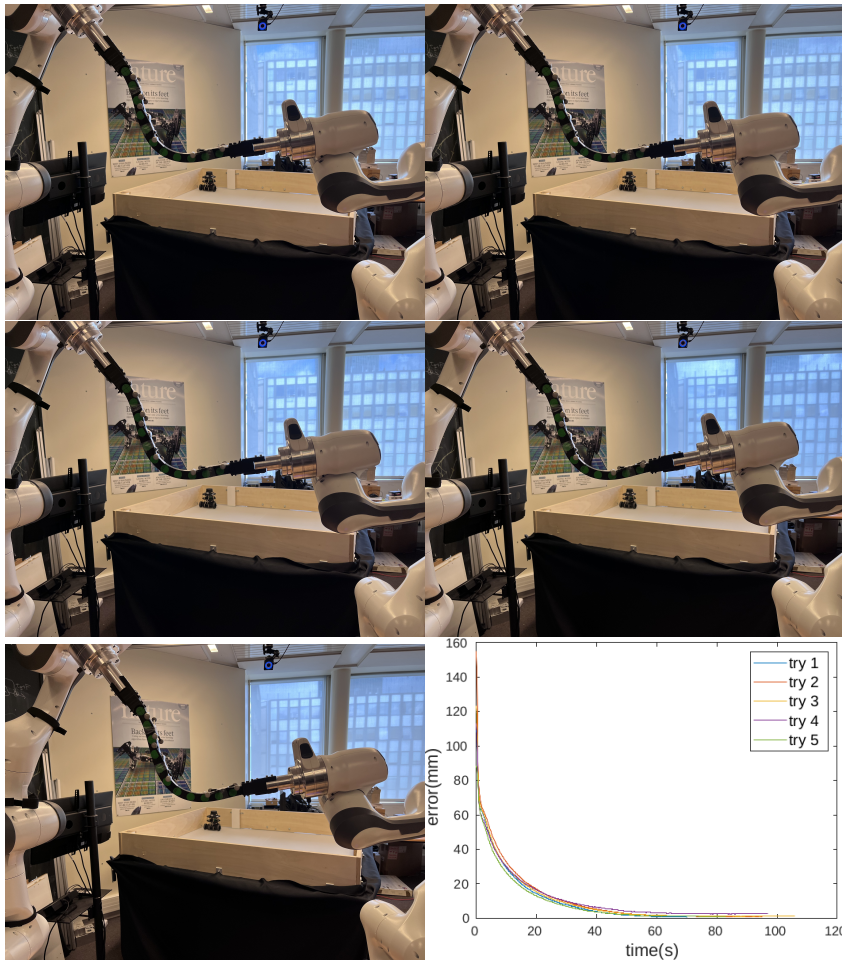


FIGURE 5.11 – Répétabilité : Formes finales pour cinq essais successifs, ainsi que l’erreur associée en fonction du temps

pas les paramètres. On a également testé la répétabilité de l’approche ainsi que, dans une certaine mesure, sa robustesse à la configuration de l’objet à l’initialisation. Les principales perspectives pour améliorer la méthode sont :

- Améliorer la robustesse à l’estimation des valeurs initiales à l’initialisation en utilisant une méthode d’estimation différente. Nous avons utilisé la méthode de tir parce qu’elle est rapide à mettre en place, mais il existe d’autres méthodes spécialement développées pour la perception et bien plus adaptées [Lilge 2022].
- Gérer les contacts, c’est un des problèmes majeurs dans la communauté de manipulation d’objets déformables. Il n’existe pas à notre connaissance, de méthode permettant de gérer des contacts non modélisés au préalable et en boucle fermée de manière efficace. La

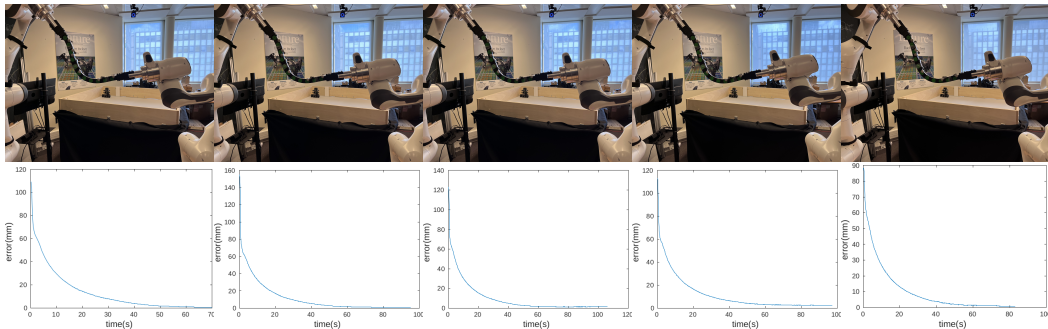


FIGURE 5.12 – Répétabilité : Formes finales pour les cinq essais successifs ainsi que l’erreur en fonction du temps associée

nature discontinue des contacts les rend extrêmement difficiles à modéliser et requiert souvent des méthodes coûteuses d’optimisation pour résoudre le problème où ils sont présents. L’utilisation du modèle Cosserat permet d’entrer des contacts dans le modèle (voir chapitre 4), mais cela implique que l’on connaît leur position sur la poutre et leur amplitude. Dans notre cas, cela requiert une méthode de perception beaucoup plus complexe qui puisse non seulement suivre la forme de l’objet, mais également détecter et identifier la direction et l’amplitude des forces appliquées par les contacts. On pourrait ainsi les prendre en compte dans la solution. Ces méthodes de perception ne sont malheureusement pas encore disponibles sur étagère.

CHAPITRE 6

Conclusion

Ce chapitre présente à la fois un récapitulatif et les conclusions de cette thèse, mais également les limites des approches proposées ainsi que les pistes de recherche que nous n'avons pas eu le temps de développer durant cette thèse.

6.1 Récapitulatif

Au cours de cette thèse, nous avons réalisé un état de l'art de la littérature sur les méthodes de modélisation d'objets déformables. Pour modéliser nos objets déformables linéaires, nous avons choisi d'utiliser un modèle basé sur la théorie de Cosserat. Nous avons choisi ce modèle pour sa précision dans les cas des objets linéaires, ainsi que pour ses bases physiques solides provenant de la mécanique continue des poutres. Ce lien physique permet notamment de modéliser des efforts extérieurs et des contacts en tout point de l'objet.

Nous avons ensuite étudié les différentes approches robotiques utilisant le modèle Cosserat. L'écrasante majorité de celles-ci ont été développées pour modéliser des robots déformables, réels ou simulés. Afin de résoudre un problème de manipulation d'objets déformables en contexte industriel, nous avons reformulé ces approches de sorte à minimiser l'erreur par rapport à la forme totale de l'objet, et non plus la position et l'orientation du bout d'un robot. Dans le cas des robots déformables, le problème aux limites est généralement formulé comme un problème de poutre encastré-manipulé. Nous avons ainsi reformulé le problème classique pour prendre en compte un ensemble plus vaste de conditions aux limites.

Ces reformulations impliquent que la solution au problème de Cosserat inverse n'est plus unique. On a donc défini un nouvel objectif à partir de la théorie de la commande optimale. Ce deuxième objectif permet ainsi de minimiser l'énergie élastique dans l'objet manipulé. Parmi les solutions qui respectent la forme imposée, on va ainsi choisir celle qui minimise les efforts appliqués par le robot et qui est donc la plus stable. En résolvant le problème aux valeurs limites (BVP) avec la méthode de tir, on trouve une configuration de l'objet optimale qui minimise à la fois l'énergie élastique dans l'objet,

et l'erreur en position. Ce critère est particulièrement important lorsqu'on manipule des objets fragiles ou qui ont une plage élastique faible tels que les plastiques ou le caoutchouc à température élevée.

Le principal défaut des algorithmes d'optimisation locaux utilisés est qu'ils sont susceptibles de tomber dans des minima locaux. Afin de s'assurer que l'on trouve une solution globale au problème, on définit un ensemble d'objectifs intermédiaires. On résout itérativement ces problèmes intermédiaires en utilisant le résultat à chaque étape comme point d'initialisation de l'algorithme à l'étape suivante. Cette méthode permet non seulement d'améliorer la recherche de solution, mais également de définir un ensemble de configurations optimales intermédiaires de l'objet. En définissant la vitesse du préhenseur de sorte à respecter les hypothèses quasi-statiques, on peut ainsi définir une trajectoire optimale de l'objet. A partir de cette trajectoire de l'objet, on fait l'hypothèse de contacts parfaits entre le robot et l'objet pour obtenir la trajectoire du robot. On applique ensuite cette méthode à une tâche de contrôle de forme d'un objet déformable linéaire.

Cette approche a toutefois des limites. Tout d'abord elle n'est pas robuste aux erreurs dans les conditions initiales de l'expérience. Par exemple, si la pose initiale de l'objet ne correspond pas à celle entrée dans le modèle, l'optimalité de la trajectoire n'est pas assurée. Ce problème est particulièrement visible lorsqu'on applique la méthode à une tâche similaire au démoulage de semelle du cas d'étude Plastinher.

Afin d'améliorer la solution et la robustesse à l'ensemble des phénomènes non modélisés, on a dans un premier temps remplacé l'algorithme d'optimisation local par un algorithme génétique global. En définissant bien la population initiale, on s'assure ainsi d'obtenir une solution globale. On peut de plus choisir une solution sur le front de Pareto obtenu en fonction des poids qu'on attribue à chacun des objectifs.

Dans un second temps, on a remplacé le contrôleur en position du robot par un contrôleur en impédance. On a ensuite défini une tâche présentant des difficultés similaires au cas d'étude du démoulage de semelle. On a utilisé notre méthode pour déterminer une trajectoire permettant au robot de détacher la bande de manière optimale. La tâche comprend des contacts difficilement modélisables comme le velcro, on a résolu ce problème en introduisant de la compliance dans le système. Cela permet à l'objet de guider lui-même le préhenseur vers une configuration minimale de l'énergie élastique sans avoir à modéliser précisément des contacts complexes. On a défini les gains du contrôleur de manière à ce que ce dernier reste rigide dans

le plan de la tâche tout en étant compliant dans les autres directions.

Cette méthode nous a jusqu'ici permis de définir une trajectoire optimale de l'objet. Toutefois, en plus d'être basée sur le modèle Cosserat et donc d'être sensible à l'estimation des paramètres visco-élastiques de l'objet, elle est calculée hors-ligne. Cela implique que l'objet va converger vers la consigne avec une erreur statique qui reflète l'ensemble des imperfections et des erreurs d'estimation du modèle. De plus, la trajectoire n'est pas recalculée en ligne et ne peut donc pas s'adapter aux perturbations au cours de la manipulation.

Afin de pallier ces problèmes, on s'est penché dans la dernière partie sur un contrôleur en boucle fermée basé sur le modèle Cosserat. Ce contrôleur peut être utilisé pour atteindre des poses générées avec la méthode précédente, ou de manière indépendante. On se base sur la linéarisation du modèle Cosserat autour d'un équilibre local pour déterminer les commandes à appliquer au robot à chaque itération. Contrairement aux approches développées dans la littérature, on recalcule la matrice Jacobienne locale à partir de l'estimation de la configuration de l'objet à chaque itération. La méthode n'est donc pas dépendante des déplacements précédents du robot. De plus, cette matrice Jacobienne crée un lien entre les variations entre les conditions initiales et les variations de position des points de contrôle. On résout ainsi le problème aux valeurs initiales après avoir mis à jour les conditions initiales spatiales de l'objet à chaque itération. Cela nous permet de nous assurer que la forme mise à jour de l'objet est physiquement réaliste, et également de modéliser d'éventuels contacts extérieurs. En bouclant le contrôleur sur le modèle et en calculant une matrice Jacobienne qui ne représente pas une pose, mais un taux de variation de la position de certains points, on s'assure ainsi que l'approche ne nécessite pas de connaissance précise des paramètres élastiques de l'objet.

On a validé l'approche pour une application de contrôle de forme en manipulant un objet linéaire avec deux bras collaboratifs Franka Emika. On a ensuite testé la répétabilité et la précision de l'objet en atteignant des erreurs inférieures à 1mm, même avec des objets avec une structure complexe tels que des câbles métalliques tressés et gainés, et en cas de glissement entre le préhenseur et l'objet durant la manipulation.

6.2 Limitations et perspectives

Dans cette partie, on discutera des principales limitations et d'éventuelles perspectives et pistes d'amélioration pour les méthodes proposées :

— **Généralisation à d'autres types d'objets** : On a vu que le

modèle Cosserat peut être étendu à d'autres géométries proches si la tâche permet de négliger les déformations dans les directions qui ne respectent pas les hypothèses du modèle. Les approches développées dans cette thèse sont néanmoins difficilement généralisables à des objets avec des géométries ou des propriétés drastiquement différentes. Les approches proposées sont directement liées à la formulation du modèle (BVP pour la planification et IVP pour le contrôle). Il serait théoriquement possible de remplacer le modèle Cosserat par un autre modèle plus général, par exemple les éléments finis, mais cela impliquerait de reformuler en profondeur les méthodes présentées ici. De plus, nous pensons qu'il n'est pas nécessaire pour une méthode d'être applicable à chaque type d'objet déformable. Ce qui importe principalement est que la méthode puisse être généralisée à l'ensemble, ou du moins à une grande partie des objets au sein d'une même catégorie.

- **Perception** : La perception est une composante essentielle dans notre boucle de contrôle. C'est à partir de la position mesurée des points de contrôle que l'on estime à chaque itération les valeurs initiales de l'objet en $s = 0$. Ces valeurs initiales sont ensuite utilisées pour calculer notre Jacobienne. A l'initialisation, c'est-à-dire en $t = 0$, on doit donc avoir une estimation précise des valeurs initiales en $s = 0$. C'est à partir de cette estimation initiale que l'on va calculer la matrice Jacobienne initiale qui va ensuite nous permettre d'estimer les valeurs initiales à l'étape suivante et ainsi de suite. Obtenir cette estimation n'est pas trivial. Dans notre cas, nous l'obtenons à l'initialisation à l'aide de la méthode de tir. Il existe toutefois des méthodes plus robustes et plus fiables construites spécifiquement pour cela. L'approche est modulable, permettant d'améliorer les résultats en remplaçant le module perception par des approches plus performantes. Dans [Diezinger 2022], les auteurs ont par exemple estimé les efforts aux limites d'un objet déformable en fonction de la courbure de l'objet.
- **Gestion des contacts** : C'est l'un des principaux problèmes ouverts pour la manipulation robotique d'objets déformables. Comment adapter les commandes du robot pour atteindre la forme souhaitée en présence de contacts non prévus et non modélisés ? L'utilisation de modèles physiques permet d'ouvrir une piste dans ce sens. Équiper l'objet de capteurs de forces permettrait ainsi de mesurer l'amplitude et le point d'application des forces produites par le contact. Celles-ci seraient ainsi entrées dans le modèle. Ce n'est toutefois pas une solution

pratique. En munissant les deux bras robotiques de capteurs de force similaires à ceux présentés dans le chapitre 4, on pourrait connaître les efforts aux limites de l'objet au cours de la manipulation. Il serait ainsi possible de résoudre un problème d'optimisation similaire à celui présenté au chapitre 3. On définirait comme objectif la position mesurée de l'objet, et comme variables d'optimisation l'amplitude et le point d'application des forces extérieures dues aux contacts. En utilisant la vision, on pourrait de plus réduire la dimension du problème en détectant le point d'application, on aurait ainsi comme seules variables d'optimisation l'amplitude des efforts dans chaque direction.

En conclusion, on a proposé dans cette thèse notre approche pour résoudre des tâches de manipulation d'objets déformables linéaires. Plus généralement, nous avons tenté à travers ce manuscrit et cette thèse, d'apporter notre pierre à l'édifice dans le domaine de la manipulation robotique.

Bibliographie

- [Adagolodjo 2021] Yinoussa Adagolodjo, Federico Renda et Christian Duriez. *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, vol. 6, no. 2, pages 3910–3917, 2021. (Cité en pages 19 et 24.)
- [Aghajanzadeh 2022] Omid Aghajanzadeh, Guillaume Picard, Juan Antonio Corrales Ramon, Christophe Cariou, Roland Lenain et Youcef Mezouar. *Optimal deformation control framework for elastic linear objects*. In 2022 IEEE 18th International Conference on Automation Science and Engineering (CASE), pages 722–728. IEEE, 2022. (Cité en page 81.)
- [Alambeigi 2018] Farshid Alambeigi, Zerui Wang, Rachel Hegeman, Yun-Hui Liu et Mehran Armand. *Autonomous data-driven manipulation of unknown anisotropic deformable tissues using unmodelled continuum manipulators*. IEEE Robotics and Automation Letters, vol. 4, no. 2, pages 254–261, 2018. (Cité en page 80.)
- [Alqumsan 2019] Ahmad Abu Alqumsan, Suiyang Khoo et Michael Norton. *Robust control of continuum robots using Cosserat rod theory*. Mechanism and Machine Theory, vol. 131, pages 48–61, 2019. (Cité en page 23.)
- [Amanov 2021] Ernar Amanov, Thien-Dang Nguyen et Jessica Burgner-Kahrs. *Tendon-driven continuum robots with extensible sections—A model-based evaluation of path-following motions*. The International Journal of Robotics Research, vol. 40, no. 1, pages 7–23, 2021. (Cité en page 24.)
- [Antman 1976] Stuart S Antman et Russell C Browne. *Applications of Methods of Functional Analysis to Problems in Mechanics : Local invertibility conditions for geometrically exact nonlinear rod and shell theories, volume 503 of Lecture Notes in Mathematics*, 1976. (Cité en page 30.)
- [Antman 2005] Stuart S Antman. *Problems in nonlinear elasticity*. Nonlinear Problems of Elasticity, pages 513–584, 2005. (Cité en pages 30, 33 et 87.)
- [Arriola-Rios 2020] Veronica E Arriola-Rios, Puren Guler, Fanny Ficuciello, Danica Kragic, Bruno Siciliano et Jeremy L Wyatt. *Modeling of*

- deformable objects for robotic manipulation : A tutorial and review.* Frontiers in Robotics and AI, vol. 7, page 82, 2020. (Cité en page 11.)
- [Autumn 2006] Kellar Autumn, A Dittmore, Daniel Santos, Matthew Spenko et M Cutkosky. *Frictional adhesion : a new angle on gecko attachment.* Journal of Experimental Biology, vol. 209, no. 18, pages 3569–3579, 2006. (Cité en page 70.)
- [Balaguer 2011] Benjamin Balaguer et Stefano Carpin. *Combining imitation and reinforcement learning to fold deformable planar objects.* In 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 1405–1412. IEEE, 2011. (Cité en page 22.)
- [Battaglia 2016] Peter Battaglia, Razvan Pascanu, Matthew Lai, Danilo Jimenez Rezende et al. *Interaction networks for learning about objects, relations and physics.* Advances in neural information processing systems, vol. 29, 2016. (Cité en page 21.)
- [Bender 2015] Jan Bender, Matthias Müller et Miles Macklin. *Position-Based Simulation Methods in Computer Graphics.* In Eurographics (tutorials), page 8, 2015. (Cité en page 16.)
- [Bergou 2008] Miklós Bergou, Max Wardetzky, Stephen Robinson, Basile Audoly et Eitan Grinspun. *Discrete elastic rods.* In ACM SIGGRAPH 2008 papers, pages 1–12. 2008. (Cité en page 27.)
- [Berthold 2021] Rebecca Berthold, Max Niklas Bartholdt, Mats Wiese, Stephanie Kahms, Svenja Spindeldreier et Annika Raatz. *A preliminary study of soft material robotic modelling : Finite element method and cosserat rod model.* In 2021 9th International Conference on Control, Mechatronics and Automation (ICCMA), pages 7–13. IEEE, 2021. (Cité en page 20.)
- [Bicchi 1995] Antonio Bicchi. *On the closure properties of robotic grasping.* The International Journal of Robotics Research, vol. 14, no. 4, pages 319–334, 1995. (Cité en page 2.)
- [Black 2017] Caroline B Black, John Till et D Caleb Rucker. *Parallel continuum robots : Modeling, analysis, and actuation-based force sensing.* IEEE Transactions on Robotics, vol. 34, no. 1, pages 29–47, 2017. (Cité en page 24.)
- [Boyer 2020] Frédéric Boyer, Vincent Lebastard, Fabien Candelier et Federico Renda. *Dynamics of continuum and soft robots : A strain parameterization based approach.* IEEE Transactions on Robotics, vol. 37, no. 3, pages 847–863, 2020. (Cité en pages 27 et 29.)
- [Boyer 2022] Frédéric Boyer, Vincent Lebastard, Fabien Candelier, Federico Renda et Mazen Alamir. *Statics and dynamics of continuum robots*

- based on Cosserat rods and optimal control theories*. IEEE Transactions on Robotics, vol. 39, no. 2, pages 1544–1562, 2022. (Cité en pages 28, 30, 37, 38 et 42.)
- [Brockman 2016] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang et Wojciech Zaremba. *Openai gym*. arXiv preprint arXiv :1606.01540, 2016. (Cité en page 22.)
- [Bryson 2014] Caroline E Bryson et D Caleb Rucker. *Toward parallel continuum manipulators*. In 2014 IEEE International Conference on Robotics and Automation (ICRA), pages 778–785. IEEE, 2014. (Cité en page 24.)
- [Campisano 2021] Federico Campisano, Simone Caló, Andria A Ramirez, James H Chandler, Keith L Obstein, Robert J Webster III et Pietro Valdastri. *Closed-loop control of soft continuum manipulators under tip follower actuation*. The International journal of robotics research, vol. 40, no. 6-7, pages 923–938, 2021. (Cité en pages 19, 25, 26, 30, 37, 48, 79, 80, 88 et 89.)
- [Chaumette 2006] François Chaumette et Seth Hutchinson. *Visual servo control. I. Basic approaches*. IEEE Robotics & Automation Magazine, vol. 13, no. 4, pages 82–90, 2006. (Cité en pages 40, 85, 87 et 88.)
- [Chenevier 2018] Jean Chenevier, David González, J Vicente Aguado, Francisco Chinesta et Elías Cueto. *Reduced-order modeling of soft robots*. PloS one, vol. 13, no. 2, page e0192052, 2018. (Cité en page 18.)
- [Chirikjian 1993] Gregory S Chirikjian. *A continuum approach to hyper-redundant manipulator dynamics*. In Proceedings of 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'93), volume 2, pages 1059–1066. IEEE, 1993. (Cité en page 19.)
- [Clegg 2018] Alexander Clegg, Wenhao Yu, Jie Tan, C Karen Liu et Greg Turk. *Learning to dress : Synthesizing human dressing motion via deep reinforcement learning*. ACM Transactions on Graphics (TOG), vol. 37, no. 6, pages 1–10, 2018. (Cité en page 22.)
- [Cocuzza 2018] Silvio Cocuzza et X-T Yan. *First engineering framework for the out-of-plane robotic shaping of thin rheological objects*. Robotics and Computer-Integrated Manufacturing, vol. 53, pages 108–121, 2018. (Cité en page 15.)
- [Coevoet 2017] Eulalie Coevoet, Thor Morales-Bieze, Frederick Largilliere, Zhongkai Zhang, Maxime Thieffry, Mario Sanz-Lopez, Bruno Carrez, Damien Marchal, Olivier Goury, Jeremie Dequid *et al.* *Software toolkit for modeling, simulation, and control of soft robots*. Advanced

- Robotics, vol. 31, no. 22, pages 1208–1224, 2017. (Cit  en pages 19 et 82.)
- [Coevoet 2019] Eulalie Coevoet, Adrien Escande et Christian Duriez. *Soft robots locomotion and manipulation control using FEM simulation and quadratic programming*. In 2019 2nd IEEE International Conference on Soft Robotics (RoboSoft), pages 739–745. IEEE, 2019. (Cit  en pages 19, 37 et 82.)
- [Cretu 2011] Ana-Maria Cretu, Pierre Payeur et Emil M Petriu. *Soft object deformation monitoring and learning for model-based robotic hand manipulation*. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 42, no. 3, pages 740–753, 2011. (Cit  en page 21.)
- [Das 2011] Jadav Das et Nilanjan Sarkar. *Autonomous shape control of a deformable object by multiple manipulators*. Journal of Intelligent & Robotic Systems, vol. 62, pages 3–27, 2011. (Cit  en page 15.)
- [Diezinger 2022] Matyas A Diezinger, Brahim Tamadazte et Guillaume J Laurent. *3d curvature-based tip load estimation for continuum robots*. IEEE Robotics and Automation Letters, vol. 7, no. 4, pages 10526–10533, 2022. (Cit  en page 108.)
- [Dou 2017] Mingsong Dou, Philip Davidson, Sean Ryan Fanello, Sameh Khamis, Adarsh Kowdle, Christoph Rhemann, Vladimir Tankovich et Shahram Izadi. *Motion2fusion : Real-time volumetric performance capture*. ACM Transactions on Graphics (ToG), vol. 36, no. 6, pages 1–16, 2017. (Cit  en page 17.)
- [Duenser 2018] Simon Duenser, James M Bern, Roi Poranne et Stelian Coros. *Interactive robotic manipulation of elastic objects*. In 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 3476–3481. IEEE, 2018. (Cit  en pages 19 et 42.)
- [Dupont 2009] Pierre E Dupont, Jesse Lock, Brandon Itkowitz et Evan Butler. *Design and control of concentric-tube robots*. IEEE Transactions on Robotics, vol. 26, no. 2, pages 209–225, 2009. (Cit  en page 23.)
- [Duriez 2013] Christian Duriez. *Control of elastic soft robots based on real-time finite element method*. In 2013 IEEE international conference on robotics and automation, pages 3982–3987. IEEE, 2013. (Cit  en page 19.)
- [Duriez 2016] Christian Duriez, Eulalie Coevoet, Fr d rick Largilliere, T Morales-Bieze, Zhongkai Zhang, Mario Sanz-Lopez, Bruno Carrez, Damien Marchal, Olivier Goury et J r mie Dequidt. *Framework for online simulation of soft robots with optimization-based inverse model*.

- In 2016 IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAN), pages 111–118. IEEE, 2016. (Cité en pages 18, 19 et 40.)
- [Edelmann 2017] Janis Edelmann, Andrew J Petruska et Bradley J Nelson. *Magnetic control of continuum devices*. The International Journal of Robotics Research, vol. 36, no. 1, pages 68–85, 2017. (Cité en pages 24 et 88.)
- [Fallahi 2016] Bitah Fallahi, Carlos Rossa, Ron S Sloboda, Nawaid Usmani et Mahdi Tavakoli. *Sliding-based switching control for image-guided needle steering in soft tissue*. IEEE Robotics and Automation Letters, vol. 1, no. 2, pages 860–867, 2016. (Cité en page 24.)
- [Faure 2012] François Faure, Christian Duriez, Hervé Delingette, Jérémie Allard, Benjamin Gilles, Stéphanie Marchesseau, Hugo Talbot, Hadrien Courtecuisse, Guillaume Bousquet, Igor Peterlik *et al.* *Sofa : A multi-model framework for interactive physical simulation*. In Soft tissue biomechanical modeling for computer assisted surgery, pages 283–321. Springer, 2012. (Cité en page 18.)
- [Ficuciello 2018] Fanny Ficuciello, Alessandro Migliozi, Eulalie Coevoet, Antoine Petit et Christian Duriez. *FEM-based deformation control for dexterous manipulation of 3D soft objects*. In 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 4007–4013. IEEE, 2018. (Cité en pages 18, 19 et 40.)
- [Frank 2010] Barbara Frank, Rüdiger Schmedding, Cyrill Stachniss, Matthias Teschner et Wolfram Burgard. *Learning the elasticity parameters of deformable objects with a manipulation robot*. In 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 1877–1883. IEEE, 2010. (Cité en page 21.)
- [Fugl 2012] Andreas Rune Fugl, Andreas Jordt, Henrik Gordon Petersen, Morten Willatzen et Reinhard Koch. *Simultaneous estimation of material properties and pose for deformable objects from depth and color images*. In Joint DAGM (German Association for Pattern Recognition) and OAGM Symposium, pages 165–174. Springer, 2012. (Cité en page 21.)
- [Gazzola 2018] Mattia Gazzola, LH Dudte, AG McCormick et Lakshminarayanan Mahadevan. *Forward and inverse problems in the mechanics of soft filaments*. Royal Society open science, vol. 5, no. 6, page 171628, 2018. (Cité en page 24.)
- [Gilbert 2016] Hunter B Gilbert, D Caleb Rucker et Robert J Webster III. *Concentric tube robots : The state of the art and future directions*. In

- Robotics Research : The 16th International Symposium ISRR, pages 253–269. Springer, 2016. (Cit  en page 88.)
- [Giraud 2022] Victor H Giraud, Maxime Padrin, Mohammadreza Shetab-Bushehri, Chedli Bouzgarrou, Youcef Mezouar et Erol Ozgur. *Optimal Shape Servoing with Task-focused Convergence Constraints*. In 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 2197–2202. IEEE, 2022. (Cit  en pages 81 et 82.)
- [Goury 2018] Olivier Goury et Christian Duriez. *Fast, generic, and reliable control and simulation of soft robots using model order reduction*. IEEE Transactions on Robotics, vol. 34, no. 6, pages 1565–1576, 2018. (Cit  en pages 18 et 19.)
- [Gravagne 2003] Ian A Gravagne, Christopher D Rahn et Ian D Walker. *Large deflection dynamics and control for planar continuum robots*. IEEE/ASME transactions on mechatronics, vol. 8, no. 2, pages 299–307, 2003. (Cit  en page 19.)
- [Grazioso 2019] Stanislao Grazioso, Giuseppe Di Gironimo et Bruno Siciliano. *A geometrically exact model for soft continuum robots : The finite element deformation space formulation*. Soft robotics, vol. 6, no. 6, pages 790–811, 2019. (Cit  en page 19.)
- [Guler 2015] Puren Guler, Karl Pauwels, Alessandro Pieropan, Hedvig Kjellstr m et Danica Kragic. *Estimating the deformability of elastic materials using optical flow and position-based dynamics*. In 2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids), pages 965–971. IEEE, 2015. (Cit  en page 16.)
- [G ler 2017] P ren G ler, Alessandro Pieropan, Masatoshi Ishikawa et Danica Kragic. *Estimating deformability of objects using meshless shape matching*. In 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 5941–5948. IEEE, 2017. (Cit  en page 16.)
- [Guo 2013] Feng Guo, Huan Lin et Yan-Bin Jia. *Squeeze grasping of deformable planar objects with segment contacts and stick/slip transitions*. In 2013 IEEE International Conference on Robotics and Automation, pages 3736–3741. IEEE, 2013. (Cit  en page 2.)
- [Ha 2015] Junhyoung Ha, Frank C Park et Pierre E Dupont. *Elastic stability of concentric tube robots subject to external loads*. IEEE Transactions on Biomedical Engineering, vol. 63, no. 6, pages 1116–1128, 2015. (Cit  en page 28.)
- [Ha 2018] Junhyoung Ha, Georgios Fagogenis et Pierre E Dupont. *Modeling tube clearance and bounding the effect of friction in concentric tube*

- robot kinematics*. IEEE Transactions on Robotics, vol. 35, no. 2, pages 353–370, 2018. (Cité en page 28.)
- [Herguedas 2019] Rafael Herguedas, Gonzalo López-Nicolás, Rosario Aragués et Carlos Sagüés. *Survey on multi-robot manipulation of deformable objects*. In 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), pages 977–984. IEEE, 2019. (Cité en page 11.)
- [Hirai 2001] Shinichi Hirai, Tatsuhiko Tsuboi et Takahiro Wada. *Robust grasping manipulation of deformable objects*. In Proceedings of the 2001 IEEE International Symposium on Assembly and Task Planning (ISATP2001). Assembly and Disassembly in the Twenty-first Century.(Cat. No. 01TH8560), pages 411–416. IEEE, 2001. (Cité en page 14.)
- [Holden 2019] Daniel Holden, Bang Chi Duong, Sayantan Datta et Derek Nowrouzezahrai. *Subspace neural physics : Fast data-driven interactive simulation*. In Proceedings of the 18th annual ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pages 1–12, 2019. (Cité en page 21.)
- [Hsiao 1999] Kuo Mo Hsiao, Jer Yan Lin et Wen Yi Lin. *A consistent co-rotational finite element formulation for geometrically nonlinear dynamic analysis of 3-D beams*. Computer Methods in Applied Mechanics and Engineering, vol. 169, no. 1-2, pages 1–18, 1999. (Cité en page 18.)
- [Hu 2018] Zhe Hu, Peigen Sun et Jia Pan. *Three-dimensional deformable object manipulation using fast online gaussian process regression*. IEEE Robotics and Automation Letters, vol. 3, no. 2, pages 979–986, 2018. (Cité en page 80.)
- [Ikawa 2007] Takeshi Ikawa et Hiroshi Noborio. *On the precision and efficiency of hierarchical rheology MSD model*. In 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 376–383. IEEE, 2007. (Cité en page 15.)
- [Janabi-Sharifi 2021] Farrokh Janabi-Sharifi, Amir Jalali et Ian D Walker. *Cosserat rod-based dynamic modeling of tendon-driven continuum robots : A tutorial*. IEEE Access, vol. 9, pages 68703–68719, 2021. (Cité en page 24.)
- [Jia 2011] Yan-Bin Jia, Feng Guo et Jiang Tian. *On two-finger grasping of deformable planar objects*. In 2011 IEEE international conference on robotics and automation, pages 5261–5266. IEEE, 2011. (Cité en page 2.)

- [Jia 2013] Yan-Bin Jia, Huan Lin et Feng Guo. *Optimal two-finger squeezing of deformable objects*. In 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 3514–3519. IEEE, 2013. (Cité en page 2.)
- [Jiménez 2012] P Jiménez. *Survey on model-based manipulation planning of deformable objects*. Robotics and computer-integrated manufacturing, vol. 28, no. 2, pages 154–163, 2012. (Cité en page 11.)
- [Jones 2009] Bryan A Jones, Ricky L Gray et Krishna Turlapati. *Three dimensional statics for continuum robotics*. In 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 2659–2664. IEEE, 2009. (Cité en page 23.)
- [Jurdjevic 1995] Velimir Jurdjevic. *Non-euclidean elastica*. American Journal of Mathematics, vol. 117, no. 1, pages 93–124, 1995. (Cité en page 28.)
- [Jurdjevic 1997] Velimir Jurdjevic. *Geometric control theory*. Cambridge university press, 1997. (Cité en page 28.)
- [Kerschen 2009] Gaëtan Kerschen, Maxime Peeters, Jean-Claude Golinval et Alexander F Vakakis. *Nonlinear normal modes, Part I : A useful framework for the structural dynamicist*. Mechanical systems and signal processing, vol. 23, no. 1, pages 170–194, 2009. (Cité en page 18.)
- [Khadem 2016] Mohsen Khadem, Carlos Rossa, Ron S Sloboda, Nawaid Usmani et Mahdi Tavakoli. *Ultrasound-guided model predictive control of needle steering in biological tissue*. Journal of Medical Robotics Research, vol. 1, no. 01, page 1640007, 2016. (Cité en page 24.)
- [Koessler 2021] Adrien Koessler, N Roca Filella, Belhassen-Chedli Bouzgarrou, Laurent Lequière et J-A Corrales Ramon. *An efficient approach to closed-loop shape control of deformable objects using finite element models*. In 2021 IEEE International Conference on Robotics and Automation (ICRA), pages 1637–1643. IEEE, 2021. (Cité en pages 19, 40 et 80.)
- [Kratchman 2016] Louis B Kratchman, Trevor L Bruns, Jake J Abbott et Robert J Webster. *Guiding elastic rods with a robot-manipulated magnet for medical applications*. IEEE Transactions on Robotics, vol. 33, no. 1, pages 227–233, 2016. (Cité en page 24.)
- [Kudoh 2015] Shunsuke Kudoh, Tomoyuki Gomi, Ryota Katano, Tetsuo Tomizawa et Takashi Suehiro. *In-air knotting of rope by a dual-arm multi-finger robot*. In 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 6202–6207. IEEE, 2015. (Cité en page 21.)

- [Lagneau 2020] Romain Lagneau, Alexandre Krupa et Maud Marchal. *Automatic shape control of deformable wires based on model-free visual servoing*. IEEE Robotics and Automation Letters, vol. 5, no. 4, pages 5252–5259, 2020. (Cité en page 80.)
- [Lang 2011] Holger Lang, Joachim Linn et Martin Arnold. *Multi-body dynamics simulation of geometrically exact Cosserat rods*. Multibody System Dynamics, vol. 25, no. 3, pages 285–312, 2011. (Cité en pages 23 et 24.)
- [Lax 1958] Peter Lax et Burton Wendroff. *Systems of conservation laws*. Rapport technique, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 1958. (Cité en page 23.)
- [Lee 2015] Alex X Lee, Henry Lu, Abhishek Gupta, Sergey Levine et Pieter Abbeel. *Learning force-based manipulation of deformable objects from multiple demonstrations*. In 2015 IEEE International Conference on Robotics and Automation (ICRA), pages 177–184. IEEE, 2015. (Cité en page 21.)
- [Leizea 2015] Ibai Leizea, Ainitze Mendizabal, Hugo Alvarez, Iker Aguinaga, Diego Borro et Emilio Sanchez. *Real-time visual tracking of deformable objects in robot-assisted surgery*. IEEE computer graphics and applications, vol. 37, no. 1, pages 56–68, 2015. (Cité en page 14.)
- [Levi 2014] Zohar Levi et Craig Gotsman. *Smooth rotation enhanced as-rigid-as-possible mesh animation*. IEEE transactions on visualization and computer graphics, vol. 21, no. 2, pages 264–277, 2014. (Cité en page 17.)
- [Lilge 2022] Sven Lilge, Timothy D Barfoot et Jessica Burgner-Kahrs. *Continuum robot state estimation using Gaussian process regression on SE (3)*. The International Journal of Robotics Research, page 02783649221128843, 2022. (Cité en page 102.)
- [Lin 2021] Xingyu Lin, Yufei Wang, Jake Olkin et David Held. *Softgym : Benchmarking deep reinforcement learning for deformable object manipulation*. In Conference on Robot Learning, pages 432–448. PMLR, 2021. (Cité en page 22.)
- [Lippi 2020] Martina Lippi, Petra Poklukar, Michael C Welle, Anastasiia Varava, Hang Yin, Alessandro Marino et Danica Kragic. *Latent space roadmap for visual action planning of deformable and rigid object manipulation*. In 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 5619–5626. IEEE, 2020. (Cité en page 21.)

- [Macchelli 2009] Alessandro Macchelli, Claudio Melchiorri et Stefano Stramigioli. *Port-based modeling and simulation of mechanical systems with rigid and flexible links*. IEEE transactions on robotics, vol. 25, no. 5, pages 1016–1029, 2009. (Cité en page 19.)
- [Macklin 2014] Miles Macklin, Matthias Müller, Nuttapong Chentanez et Tae-Yong Kim. *Unified particle physics for real-time applications*. ACM Transactions on Graphics (TOG), vol. 33, no. 4, pages 1–12, 2014. (Cité en page 16.)
- [Macklin 2016] Miles Macklin, Matthias Müller et Nuttapong Chentanez. *XPBD : position-based simulation of compliant constrained dynamics*. In Proceedings of the 9th International Conference on Motion in Games, pages 49–54, 2016. (Cité en page 16.)
- [Matas 2018] Jan Matas, Stephen James et Andrew J Davison. *Sim-to-real reinforcement learning for deformable object manipulation*. In Conference on Robot Learning, pages 734–743. PMLR, 2018. (Cité en page 22.)
- [Mordatch 2012] Igor Mordatch, Emanuel Todorov et Zoran Popović. *Discovery of complex behaviors through contact-invariant optimization*. ACM Transactions on Graphics (ToG), vol. 31, no. 4, pages 1–8, 2012. (Cité en page 22.)
- [Moré 2006] Jorge J Moré. *The Levenberg-Marquardt algorithm : implementation and theory*. In Numerical analysis : proceedings of the biennial Conference held at Dundee, June 28–July 1, 1977, pages 105–116. Springer, 2006. (Cité en page 46.)
- [Müller 2008] Matthias Müller, Jos Stam, Doug James et Nils Thürey. *Real time physics : class notes*. In ACM siggraph 2008 classes, pages 1–90. 2008. (Cité en page 14.)
- [Murray 1994] Richard M Murray, Zexiang Li, S Shankar Sastry et S Shankara Sastry. *A mathematical introduction to robotic manipulation*. CRC press, 1994. (Cité en pages 30 et 31.)
- [Navarro-Alarcon 2013] David Navarro-Alarcon, Yun-Hui Liu, José Guadalupe Romero et Peng Li. *Model-free visually servoed deformation control of elastic objects by robot manipulators*. IEEE Transactions on Robotics, vol. 29, no. 6, pages 1457–1468, 2013. (Cité en page 80.)
- [Navarro-Alarcon 2014] David Navarro-Alarcon, Yun-hui Liu, Jose Guadalupe Romero et Peng Li. *On the visual deformation servoing of compliant objects : Uncalibrated control methods and experiments*. The

- International Journal of Robotics Research, vol. 33, no. 11, pages 1462–1480, 2014. (Cité en pages 40 et 80.)
- [Navarro-Alarcon 2016] David Navarro-Alarcon, Hiu Man Yip, Zerui Wang, Yun-Hui Liu, Fangxun Zhong, Tianxue Zhang et Peng Li. *Automatic 3-d manipulation of soft objects by robotic arms with an adaptive deformation model*. IEEE Transactions on Robotics, vol. 32, no. 2, pages 429–441, 2016. (Cité en page 80.)
- [Navarro-Alarcon 2017] David Navarro-Alarcon et Yun-Hui Liu. *Fourier-based shape servoing : A new feedback method to actively deform soft objects into desired 2-D image contours*. IEEE Transactions on Robotics, vol. 34, no. 1, pages 272–279, 2017. (Cité en page 80.)
- [Nocedal 2006] Jorge Nocedal et Stephen J Wright. *Quadratic programming*. Numerical optimization, pages 448–492, 2006. (Cité en page 46.)
- [Oliver-Butler 2019] Kaitlin Oliver-Butler, John Till et Caleb Rucker. *Continuum robot stiffness under external loads and prescribed tendon displacements*. IEEE Transactions on Robotics, vol. 35, no. 2, pages 403–419, 2019. (Cité en page 24.)
- [Orekhov 2020] Andrew L Orekhov et Nabil Simaan. *Solving cosserat rod models via collocation and the magnus expansion*. In 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 8653–8660. IEEE, 2020. (Cité en page 24.)
- [O’Reilly 2011] Oliver M O’Reilly et Daniel M Peters. *On stability analyses of three classical buckling problems for the elastic strut*. Journal of Elasticity, vol. 105, pages 117–136, 2011. (Cité en pages 50, 89 et 100.)
- [Parashar 2015] Shaifali Parashar, Daniel Pizarro, Adrien Bartoli et Toby Collins. *As-rigid-as-possible volumetric shape-from-template*. In Proceedings of the IEEE International Conference on Computer Vision, pages 891–899, 2015. (Cité en page 17.)
- [Petit 2017] Antoine Petit, Fanny Ficuciello, Giuseppe Andrea Fontanelli, Luigi Villani et Bruno Siciliano. *Using physical modeling and RGB-D registration for contact force sensing on deformable objects*. In ICINCO 2017-14th International Conference on Informatics in Control, Automation and Robotics, volume 2, pages 24–33. ScitePress ; Springer, 2017. (Cité en page 21.)
- [Petit 2018] Antoine Petit, Stéphane Cotin, Vincenzo Lippiello et Bruno Siciliano. *Capturing deformations of interacting non-rigid objects using rgb-d data*. In 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 491–497. IEEE, 2018. (Cité en page 21.)

- [Pignat 2017] Emmanuel Pignat et Sylvain Calinon. *Learning adaptive dressing assistance from human demonstration*. Robotics and Autonomous Systems, vol. 93, pages 61–75, 2017. (Cité en page 22.)
- [Potra 2000] Florian A Potra et Stephen J Wright. *Interior-point methods*. Journal of computational and applied mathematics, vol. 124, no. 1-2, pages 281–302, 2000. (Cité en page 46.)
- [Qu 2004] Zu-Qing Qu. *Model order reduction techniques with applications in finite element analysis : with applications in finite element analysis*. Springer Science & Business Media, 2004. (Cité en page 18.)
- [Rambow 2012] Matthias Rambow, Thomas Schauß, Martin Buss et Sandra Hirche. *Autonomous manipulation of deformable objects based on teleoperated demonstrations*. In 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 2809–2814. IEEE, 2012. (Cité en page 22.)
- [Rao 2009] Anil V Rao. *A survey of numerical methods for optimal control*. Advances in the astronautical Sciences, vol. 135, no. 1, pages 497–528, 2009. (Cité en pages 23, 28 et 45.)
- [Renda 2014] Federico Renda, Michele Giorelli, Marcello Calisti, Matteo Cianchetti et Cecilia Laschi. *Dynamic model of a multibending soft robot arm driven by cables*. IEEE Transactions on Robotics, vol. 30, no. 5, pages 1109–1122, 2014. (Cité en page 23.)
- [Renda 2020] Federico Renda, Costanza Armanini, Vincent Lebastard, Fabien Candelier et Frederic Boyer. *A geometric variable-strain approach for static modeling of soft manipulators with tendon and fluidic actuation*. IEEE Robotics and Automation Letters, vol. 5, no. 3, pages 4006–4013, 2020. (Cité en pages 24 et 29.)
- [Renda 2021] Federico Renda, Conor Messer, Caleb Rucker et Frederic Boyer. *A sliding-rod variable-strain model for concentric tube robots*. IEEE Robotics and Automation Letters, vol. 6, no. 2, pages 3451–3458, 2021. (Cité en page 29.)
- [Riahi 2009] Azadeh Riahi et John H Curran. *Full 3D finite element Cosserat formulation with application in layered structures*. Applied mathematical modelling, vol. 33, no. 8, pages 3450–3464, 2009. (Cité en page 20.)
- [Riahi 2010] A Riahi et JH Curran. *Comparison of the Cosserat Continuum approach with finite element interface models in a simulation of layered materials*. 2010. (Cité en page 20.)
- [Rucker 2010] D Caleb Rucker, Bryan A Jones et Robert J Webster III. *A geometrically exact model for externally loaded concentric-tube*

- continuum robots*. IEEE transactions on robotics, vol. 26, no. 5, pages 769–780, 2010. (Cit  en pages 23, 24, 30, 32, 37, 38, 40 et 89.)
- [Rucker 2011] D Caleb Rucker et Robert J Webster III. *Statics and dynamics of continuum robots with general tendon routing and external loading*. IEEE Transactions on Robotics, vol. 27, no. 6, pages 1033–1044, 2011. (Cit  en pages 23, 27, 37 et 38.)
- [Rucker 2013] D Caleb Rucker, Jadav Das, Hunter B Gilbert, Philip J Swaney, Michael I Miga, Nilanjan Sarkar et Robert J Webster. *Sliding mode control of steerable needles*. IEEE Transactions on Robotics, vol. 29, no. 5, pages 1289–1299, 2013. (Cit  en pages 24 et 38.)
- [Sakamoto 2009] Naoki Sakamoto, Mitsuru Higashimori, Toshio Tsuji et Makoto Kaneko. *An optimum design of robotic food handling by using Burger model*. Intelligent Service Robotics, vol. 2, pages 53–60, 2009. (Cit  en page 15.)
- [Sanchez 2018] Jose Sanchez, Juan-Antonio Corrales, Belhassen-Chedli Bouzgarrou et Youcef Mezouar. *Robotic manipulation and sensing of deformable objects in domestic and industrial applications : a survey*. The International Journal of Robotics Research, vol. 37, no. 7, pages 688–716, 2018. (Cit  en pages 6 et 12.)
- [Schulman 2013] John Schulman, Ankush Gupta, Sibi Venkatesan, Mallory Tayson-Frederick et Pieter Abbeel. *A case study of trajectory transfer through non-rigid registration for a simplified suturing scenario*. In 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 4111–4117. IEEE, 2013. (Cit  en page 22.)
- [Schulman 2016] John Schulman, Jonathan Ho, Cameron Lee et Pieter Abbeel. *Learning from demonstrations through the use of non-rigid registration*. In Robotics Research, pages 339–354. Springer, 2016. (Cit  en page 22.)
- [Shetab-Bushehri 2022] Mohammadreza Shetab-Bushehri, Miguel Aranda, Youcef Mezouar et Erol Ozgur. *Lattice-based shape tracking and servoing of elastic objects*. arXiv preprint arXiv :2209.01832, 2022. (Cit  en pages 17, 40, 81 et 82.)
- [Simo 1985] Juan C Simo. *A finite strain beam formulation. The three-dimensional dynamic problem. Part I*. Computer methods in applied mechanics and engineering, vol. 49, no. 1, pages 55–70, 1985. (Cit  en page 27.)
- [Sin 2013] Fun Shing Sin, Daniel Schroeder et Jernej Barbi . *Vega : non-linear FEM deformable object simulator*. In Computer Graphics Forum, volume 32, pages 36–48. Wiley Online Library, 2013. (Cit  en page 18.)

- [Sorkine 2007] Olga Sorkine et Marc Alexa. *As-rigid-as-possible surface modeling*. In Symposium on Geometry processing, volume 4, pages 109–116. Citeseer, 2007. (Cité en page 17.)
- [Spillmann 2007] Jonas Spillmann et Matthias Teschner. *CoRdE : Cosserat rod elements for the dynamic simulation of one-dimensional elastic objects*. In Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation, pages 63–72, 2007. (Cité en pages 24 et 25.)
- [Starke 2017] Julia Starke, Ernar Amanov, M Taha Chikhaoui et Jessica Burgner-Kahrs. *On the merits of helical tendon routing in continuum robots*. In 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 6470–6476. IEEE, 2017. (Cité en page 24.)
- [Takens 1981] Fq Takens. *Lecture notes in mathematics*. by DA Rand and L.-S. Young Springer, Berlin, vol. 898, page 366, 1981. (Cité en page 30.)
- [Tanaka 2018] Daisuke Tanaka, Solvi Arnold et Kimitoshi Yamazaki. *Emd net : An encode-manipulate-decode network for cloth manipulation*. IEEE Robotics and Automation Letters, vol. 3, no. 3, pages 1771–1778, 2018. (Cité en page 21.)
- [Terzopoulos 1987] Demetri Terzopoulos, John Platt, Alan Barr et Kurt Fleischer. *Elastically deformable models*. In Proceedings of the 14th annual conference on Computer graphics and interactive techniques, pages 205–214, 1987. (Cité en page 17.)
- [Till 2015] John Till, Caroline E Bryson, Scotty Chung, Andrew Orekhov et D Caleb Rucker. *Efficient computation of multiple coupled Cosserat rod models for real-time simulation and control of parallel continuum manipulators*. In 2015 IEEE international conference on robotics and automation (ICRA), pages 5067–5074. IEEE, 2015. (Cité en page 24.)
- [Till 2017a] John Till et D Caleb Rucker. *Elastic rod dynamics : Validation of a real-time implicit approach*. In 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 3013–3019. IEEE, 2017. (Cité en pages 24 et 88.)
- [Till 2017b] John Till et D Caleb Rucker. *Elastic stability of cosserat rods and parallel continuum robots*. IEEE Transactions on Robotics, vol. 33, no. 3, pages 718–733, 2017. (Cité en page 28.)
- [Till 2019] John Till, Vincent Aloï et Caleb Rucker. *Real-time dynamics of soft and continuum robots based on Cosserat rod models*. The International Journal of Robotics Research, vol. 38, no. 6, pages 723–746, 2019. (Cité en pages 24 et 26.)

- [Tokumoto 2001] Shinichi Tokumoto, Shinichi Hirai et Hiromi Tanaka. *Constructing virtual rheological objects*. In Proc. World Multiconference on Systemics, Cybernetics and Infomatics, pages 106–111, 2001. (Cit  en pages 7 et 16.)
- [Trivedi 2008] Deepak Trivedi, Amir Lotfi et Christopher D Rahn. *Geometrically exact models for soft robotic manipulators*. IEEE Transactions on Robotics, vol. 24, no. 4, pages 773–780, 2008. (Cit  en pages 23, 28 et 30.)
- [Tsurumine 2019] Yoshihisa Tsurumine, Yunduan Cui, Eiji Uchibe et Takamitsu Matsubara. *Deep reinforcement learning with smooth policy update : Application to robotic cloth manipulation*. Robotics and Autonomous Systems, vol. 112, pages 72–83, 2019. (Cit  en page 22.)
- [Tummers 2023] Matthias Tummers, Vincent Lebastard, Fr d ric Boyer, Jocelyne Troccaz, Benoit Rosa et M Taha Chikhaoui. *Cosserat Rod Modeling of Continuum Robots from Newtonian and Lagrangian Perspectives*. IEEE Transactions on Robotics, 2023. (Cit  en pages 29 et 30.)
- [Twardon 2018] Lukas Twardon et Helge Ritter. *Learning to put on a knit cap in a head-centric policy space*. IEEE Robotics and Automation Letters, vol. 3, no. 2, pages 764–771, 2018. (Cit  en page 22.)
- [Valencia 2020] Angel J Valencia et Pierre Payeur. *Combining self-organizing and graph neural networks for modeling deformable objects in robotic manipulation*. Frontiers in Robotics and AI, vol. 7, page 600584, 2020. (Cit  en page 21.)
- [Van Veldhuizen 1998] David A Van Veldhuizen, Gary B Lamont *et al*. *Evolutionary computation and convergence to a pareto front*. In Late breaking papers at the genetic programming 1998 conference, pages 221–228. Citeseer, 1998. (Cit  en page 63.)
- [Wada 1998] Takahiro Wada, Shinichi Hirai et Sadao Kawamura. *Indirect simultaneous positioning operations of extensionally deformable objects*. In Proceedings. 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems. Innovations in Theory, Practice and Applications (Cat. No. 98CH36190), volume 2, pages 1333–1338. IEEE, 1998. (Cit  en page 14.)
- [Wang 2009] Zhongkui Wang et Shinichi Hirai. *Modeling and parameter identification of rheological object based on FE method and nonlinear optimization*. In 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 1968–1973. IEEE, 2009. (Cit  en page 15.)

- [Wang 2010a] Zhongkui Wang et Shinichi Hirai. *Contact modeling and parameter switching for simultaneous reproduction of rheological force and deformation*. In 2010 IEEE International Conference on Robotics and Biomimetics, pages 726–731. IEEE, 2010. (Cité en page 15.)
- [Wang 2010b] Zhongkui Wang et Shinichi Hirai. *Modeling and property estimation of Japanese sweets for their manufacturing simulation*. In 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 3536–3541. IEEE, 2010. (Cité en page 15.)
- [Wang 2021] Jiaole Wang, Joseph Peine et Pierre E Dupont. *Eccentric tube robots as multiarmed steerable sheaths*. IEEE Transactions on Robotics, vol. 38, no. 1, pages 477–490, 2021. (Cité en page 29.)
- [Yan 2020] Mengyuan Yan, Yilin Zhu, Ning Jin et Jeannette Bohg. *Self-supervised learning of state estimation for manipulating deformable linear objects*. IEEE robotics and automation letters, vol. 5, no. 2, pages 2372–2379, 2020. (Cité en page 21.)
- [Yin 2021] Hang Yin, Anastasia Varava et Danica Kragic. *Modeling, learning, perception, and control methods for deformable object manipulation*. Science Robotics, vol. 6, no. 54, page eabd8803, 2021. (Cité en page 13.)
- [Yoshida 2015] Eiichi Yoshida, Ko Ayusawa, Ixchel G Ramirez-Alpizar, Kensuke Harada, Christian Duriez et Abderrahmane Kheddar. *Simulation-based optimal motion planning for deformable object*. In 2015 IEEE international workshop on advanced robotics and its social impacts (ARSO), pages 1–6. IEEE, 2015. (Cité en page 18.)
- [Zhang 2017] Zhongkai Zhang, Thor Morales Bieze, Jérémie Dequidt, Alexandre Kruszewski et Christian Duriez. *Visual servoing control of soft robots based on finite element model*. In 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 2895–2901. IEEE, 2017. (Cité en pages 79 et 80.)
- [Zhu 2018] Jihong Zhu, Benjamin Navarro, Philippe Fraise, André Crosnier et Andrea Cherubini. *Dual-arm robotic manipulation of flexible cables*. In 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 479–484. IEEE, 2018. (Cité en page 80.)
- [Zhu 2021] Jihong Zhu, David Navarro-Alarcon, Robin Passama et Andrea Cherubini. *Vision-based manipulation of deformable and rigid objects using subspace projections of 2D contours*. Robotics and Autonomous Systems, vol. 142, page 103798, 2021. (Cité en page 80.)
- [Zhu 2022] Jihong Zhu, Andrea Cherubini, Claire Dune, David Navarro-Alarcon, Farshid Alambeigi, Dmitry Berenson, Fanny Ficuciello, Kensuke Harada, Jens Kober, Xiang Liet al. *Challenges and outlook*

in robotic manipulation of deformable objects. IEEE Robotics & Automation Magazine, vol. 29, no. 3, pages 67–77, 2022. (Cité en pages 3 et 14.)

[Zienkiewicz 2005] Olek C Zienkiewicz, Robert L Taylor et Jian Z Zhu. The finite element method : its basis and fundamentals. Elsevier, 2005. (Cité en page 30.)