



HAL
open science

Control of Linear Deformable Objects for Robotized Crops Manipulation

Omid Aghajanzadeh

► **To cite this version:**

Omid Aghajanzadeh. Control of Linear Deformable Objects for Robotized Crops Manipulation. Automatic. Université Clermont Auvergne, 2022. English. NNT : 2022UCFAC086 . tel-04092448

HAL Id: tel-04092448

<https://theses.hal.science/tel-04092448>

Submitted on 9 May 2023

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.

UNIVERSITÉ CLERMONT AUVERGNE
ÉCOLE DOCTORALE SCIENCES POUR L'INGÉNIEUR
INSTITUT PASCAL, INRAE
DOCTORAL THESIS

Control of Linear Deformable Objects for Robotized Crops Manipulation

Author:
Omid Aghajanzadeh

*pour obtenir le grade de
Docteur d'Université*

Sera soutenue le 07/12/2022 devant le jury composé de :

Tahri Omar	Rapporteur	Professeur des universités, Université de Bourgogne
Krupa Alexandre	Rapporteur	Directeur de recherche, INRIA Rennes
Bouzgarrou Belhassen-Chedli	Examineur	Professeur des universités, SIGMA Clermont-Ferrand
Aragüés Rosario	Examineur	Associate professor, Universidad de Zaragoza
Aranda Miguel	Invité	Chercheur postdoctoral, Universidad de Zaragoza
Mezouar Youcef	Directeur	Professeur des universités, SIGMA Clermont-Ferrand
Lenain Roland	Encadrant	Directeur de recherche, INRAE Clermont-Ferrand
Corrales-Ramon Juan-Antonio	Encadrant	Chercheur postdoctoral, Universidade de Santiago de Compostela
Cariou Christophe	Encadrant	Ingénieur de recherche, INRAE Clermont-Ferrand



Declaration of Authorship

I, Omid Aghajanzadeh, declare that this thesis titled, Control of Linear Deformable Objects for Robotized Crops Manipulation and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

Abstract

Control of Linear Deformable Objects for Robotized Crops Manipulation

by Omid AGHAJANZADEH

The use of robots in several essential domains (domestic, industrial, and agricultural) has increased significantly. In recent years, robots have been able to execute a variety of tasks with rigid objects. However, despite the abundance of non-rigid objects all around of us, the manipulation of **deformable objects (DO)** has received significantly less attention.

Linear deformable objects (LDOs) are one type of **DO** that are commonly present in industry and agriculture, for instance, cables and twigs of vegetables. Yet, finding a suitable technique to control this type of objects remains a significant challenge due to their high **degrees of freedom (DOF)**. Thus, the direction of this research is to develop methods to deal with the stated problem. The main application we consider is agriculture but the methods can be used in other fields as well. In agriculture, many objects can be assumed as **LDOs**, such as the branches and trunks of vegetables. Pruning, harvesting, packing vegetables and fruits, and moving plants are examples of tasks that require the manipulation of **LDOs**.

A significant challenge in manipulating these objects is to deform them toward the targets desirably, in order to perform different actions (deforming, picking, etc). A major question that may arise here is how robots can properly perform the deformation tasks with **LDOs** while a wide range of **LDOs** with different characteristics exists in the agricultural field. Another challenge is that tracking their entire shape is not possible (and not necessary) during the deformation process in some cases, while in others, it is required to track and control the entire shape. Based on the mentioned problems, this dissertation is motivated by two research questions: (1) how can robots deform an **LDO** without perceiving its entire shape? (2) how can robots deform the entire shape of the object without knowing any model parameters or offline information of the object's deformation beforehand?

To address the first question, we propose two methods to manipulate the objects without perceiving the entire shape. First, we propose an offline geometric model for controlling the shape of elastic linear objects, which develops a method to calculate the deformation Jacobian based on the object's shape at rest. This method has some limitations and cannot be used with a wide range of objects. So, in the next chapter, using an adaptive algorithm, we suggest a more generic approach for controlling an arbitrary point along the length of an object without calculating any model-based Jacobian matrix. This method can track a desired manipulation trajectory to reach the target shape. To answer the second question, first, we propose an optimal controller method by using an online estimation of a deformation Jacobian to control the objects' entire shape using a geometric algorithm. Next, we extend the adaptive methodology to control the entire shape. In this method, we do not need to have a geometric model of the object. To validate the presented methods, numerical simulations and real experiments are conducted.

Keywords. Linear deformable objects, robot manipulation, adaptive control, tracking of deformation, optimal control, agricultural robotics.

Résumé

Contrôle d'objets déformables linéaires pour la manipulation robotisée de cultures

par Omid AGHAJANZADEH

De nombreux domaines peuvent bénéficier de l'apport de la robotique pour réaliser des travaux difficiles et répétitifs, ou requérant une précision particulière. A ce titre, de nombreux robots sont aujourd'hui exploités pour exécuter une grande variété de tâches avec des objets rigides. En revanche, malgré l'abondance d'objets déformables autour de nous, la manipulation d'objets déformables (en anglais : [deformable objects](#) ou [DO](#)) a reçu beaucoup moins d'attention, notamment du fait de la nécessaire adaptation des fonctions robotiques à la variabilité des propriétés d'interactions avec l'environnement.

Les objets linéaires déformables (en anglais : [linear deformable object](#) ou [LDO](#)) sont un type de [DO](#) très présent dans l'industrie en général et l'agriculture en particulier. Pourtant, trouver une technique appropriée pour contrôler ce type d'objet reste un défi important en raison de leur nombre élevé de degrés de liberté (en anglais : [degrees of freedom](#) ou [DOF](#)). Ainsi, l'objectif de cette thèse est de développer des méthodes de contrôle pour traiter le problème d'asservissement de forme. Même si les contributions de cette thèse peuvent être utilisées dans de nombreux domaines, la principale application que nous considérons dans cette thèse est l'agriculture. En effet, dans ce contexte, de nombreux objets peuvent être considérés comme des [LDOs](#), tels que les branches ou les plantes.

Le défi important lors de la manipulation de ces objets est de pouvoir les positionner selon une forme désirée, afin de réaliser différentes actions (taile, cueillette, traitement, etc). La question qui peut se poser ici est de savoir comment les robots peuvent effectuer correctement les tâches de déformation avec des [LDO](#) alors qu'une large gamme de caractéristiques différentes existe, ainsi qu'une forte variabilité de leurs propriétés. Une autre problématique est que le suivi de leur forme complète est difficilement envisageable pendant le processus de déformation, tandis que dans certain cas, il est nécessaire de suivre et de contrôler la forme entière. Sur la base des problèmes mentionnés, cette thèse cherche à répondre à deux questions de recherche : (1) comment les robots peuvent-ils déformer un [LDO](#) sans percevoir sa forme entière ? (2) comment les robots peuvent-ils déformer la forme entière de l'objet sans connaître au préalable les caractéristiques de l'objet ?

Pour répondre à la première question, nous proposons deux méthodes pour manipuler les objets sans percevoir leur forme entière. Tout d'abord, nous proposons un modèle géométrique hors ligne pour contrôler la forme d'objets linéaires élastiques, qui développe une méthode de calcul de la jacobienne de déformation en se basant sur la forme de l'objet au repos. Cette méthode présente certaines limites et ne peut être utilisée avec une large gamme d'objets. Ainsi, à l'aide d'un algorithme adaptatif, nous proposons une approche plus générique pour contrôler un point arbitraire sur la longueur d'un objet sans avoir recours à la matrice jacobienne. Cette méthode peut suivre une trajectoire de manipulation souhaitée pour atteindre la forme cible. Pour la deuxième question, nous proposons d'abord une méthode de contrôle optimale en utilisant une estimation en ligne d'une jacobienne de déformation pour contrôler la forme entière des objets en utilisant un algorithme géométrique. Ensuite, nous étendons la méthode adaptative pour contrôler la forme entière. Dans cette méthode, nous n'avons pas besoin de disposer d'un

modèle géométrique de l'objet. Afin de valider les méthodes présentées, des simulations numériques et des expériences en réel sont menées pour évaluer leurs performances.

Mots clés. Objet linéairement déformable, manipulation robotique, contrôle adaptatif, suivi de la déformation, commande optimale, robotique agricole.

*To my lovely **Family**
and my beloved **Sahar***

Acknowledgements

Many people assisted me in reaching this stage of my life and deserve my heartfelt gratitude. Yet, there are a few individuals that I must single out for special attention.

First and foremost, I would like to express my profound gratitude to my advisers, Prof. Youcef Mezouar, Dr. Roland Lenain, Dr. Juan-Antonio Corrales-Ramon, and Dr. Christophe Cariou, for having faith in my abilities and giving me this chance to pursue my education in robotics. I am grateful for their invaluable guidance throughout my Ph.D. and would remember them for being extremely friendly and supportive. Their efforts, suggestions, and timely encouragement led to the development of this thesis.

I would like to take this opportunity to express my sincere gratitude to Dr. Miguel Aranda for all the advice and guidance he has given me throughout my Ph.D., thanks for helping me to expand my ideas and develop my simulation codes, and for serving on my doctoral committee. I would also like to thank the dear members of the jury, Prof. Omar Tahri, Dr. Alexandre Krupa, Prof. Bouzgarrou Belhassen-Chedli, and Dr. Rosario Aragüés, for having accepted to review my work. It was a great honor to be evaluated by renowned researchers and I am grateful for their insightful comments, feedback, and discussion during the defense.

I also want to express my gratitude to all the wonderful people I have had the privilege of working with over the past years at Institute Pascal, SIGMA, and INRAE. Thank you to all of my lab mates in MACCS, particularly my friends Reza, Guillaume, Maxime, and Melodie. I also wish to express my sincere thanks to the fantastic support staff of CNRS, especially Dr. Sabrina Juarez, for all the support during the past years.

I am also very grateful to my parents, Farhad and Safiyeh, for enabling me to pursue my dreams in the first place. I am appreciative of their unwavering love and devotion. I am also grateful for the unconditional love and encouragement of my beloved brother and sister, Hossein and Maryam. Nevertheless, expressing gratitude to my family in just a few words would be far from sufficient. Finally, I must provide special thanks to my dear wife, Sahar, for what she has consistently done to support and love me. Without her incredible energy, love, and inspiration, nothing would be possible.

This work was sponsored by a public grant overseen by the French National Research Agency as part of the "Investissements d'Avenir" through the IMobS3 Laboratory of Excellence (ANR-10-LABX-0016) and the IDEX-ISITE initiative CAP 20-25 (ANR-16-IDEX-0001). It is developed within the context of the project "SynCEA" ("Dynamic SYNChronization of mobile manipulators for new Environmental and Agricultural tools") of CAP 20-25 Challenge 2 (Domain: Agro Technologies).

Contents

1	Introduction	1
1.1	Context	1
1.2	Motivation	3
1.3	Contribution and outline	4
1.4	Publications	6
2	State of the art and preliminaries	9
2.1	Robotics in agriculture	9
2.2	Deformation modeling	12
2.3	State-of-the-art on deformable object manipulation	19
2.4	LDO manipulation	23
2.5	Positioning of this thesis with regards to the literature	27
3	An offline geometric method for deforming LDOs	29
3.1	Introduction	29
3.2	Our contributions	30
3.3	Problem formulation	30
3.4	Proposed LDO modeling and shape control	31
3.5	ASAP as an ARAP approximation	36
3.6	Experimental validation	37
3.7	Conclusion	46
4	Adaptive deformation control for an arbitrary point	47
4.1	Introduction	47
4.2	Our contributions	48
4.3	Problem definition	49
4.4	Controller design	50
4.5	Simulation results	53
4.6	Robot experiments	57
4.7	Conclusion	62
5	Optimal deformation control framework for LDOs	63
5.1	Our contributions	63
5.2	Problem statement	64
5.3	Desired shape optimization	65
5.4	Shape control method	66
5.5	Simulation results	68
5.6	Robotic experiments	68
5.7	Conclusion	72

6 Adaptive deformation control of the entire object	75
6.1 Introduction	75
6.2 Problem definition	76
6.3 Control scheme	78
6.4 Experimental results	83
6.5 Conclusion	87
7 Conclusion and perspectives	89
7.1 Summary	89
7.2 Limitation and future perspective	91
Bibliography	92
A Comparison of the proposed methods	109

List of Figures

1.1	DOs manipulation examples	2
1.2	Examples of manipulating LDOs	3
1.3	The schematic diagram of the present thesis that depicts the chapters in sequence.	5
2.1	Several snapshots of the position of the robotic arm with respect to the target (red target board) presented in [YPS ⁺ 21]. Red arrows show the contact points. The first row shows successful cases, and the second row shows failing ones where the end effector hits the tree.	10
2.2	Approximating the DO as presented in [TKK01].	11
2.3	Different geometric models for elastic deformations presented in [YGL ⁺ 18]. The objects' shapes were constructed using a set of geometrical nodes on their surfaces without considering any physical properties.	13
2.4	Model of an object with interconnected MSD presented in [DS11]. The mass points (m_i ($i = 1, 2, \dots$)) are inserted at the nodal points to model the object. Moreover, a three-element model (E_j ($j = 1, 2, \dots$)) is placed between the nearby mass points. E_j consist of elastic and viscosity coefficients of the non-residual deformation part (k_1 and c_1), and the viscosity coefficient in the residual deformation ¹ part (c_2).	13
2.5	A flexible rod modeled in MATLAB using a large number of meshes to make the model accurate. However, due to the extensive use of mesh, it has a very high computational cost.	14
2.6	Comparison between BEM and FEM mesh presented in [GSN04]. FEM mesh is used to model the top jaw, and BEM mesh is used to model the bottom jaw.	15
2.7	Approximating the studied object as LDO	16
2.8	The schematic diagram shows the relationship between the proposed controller methods and the adopted simulated models/real objects in this thesis. As one can see, the controllers are designed independently, and the models or objects are used to validate the methods.	16
2.9	An elastic rod in static equilibrium, held at each end by robotic grippers presented in [BM14].	17
2.10	Modeling an Armadillo using ARAP presented in [SA07].	18
2.11	A deformable foam manipulation that was modeled with a non-linear mass-spring system [ZCB ⁺ 17].	20
2.12	FEM model of the object used in [DBPC18] for a simulation-based control methodology. To model the object using FEM, some physical information about the object is needed in advance.	21
2.13	Setup used in [NAYW ⁺ 16] for the automatic manipulation of the objects.	22

2.14	The robot's performance in emulating the human demonstration for deforming the rope into various shapes was compared qualitatively and presented in [NCA ⁺ 17]. The sequence of example images supplied as input to the robot is shown in the upper row. The lower row shows the states achieved by the robot as it attempts to follow the indicated trajectory. . . .	24
2.15	Experiment with large deformations of the wire presented in [LKM20]. The target shape is shown on the left, and the final shape of the wire is presented on the right. This figure depicts an example of unsuccessful attempts in large deformations with the approach provided in [LKM20]. . .	25
2.16	A successful test with intermediary targets for reaching large deformations of the wire that is presented in [LKM20]. Some of the intermediary targets are shown in the three first pictures. The final picture shows the final configuration of the wire that reached the target.	25
2.17	An experiment performed with the approach presented in [QMNA ⁺ 20]. The target is shown in red, and the green curves illustrate the path that the object took to get there from the initial shape.	26
2.18	An experiment performed with the method proposed by [CCR21]. The image on the left shows the initial configuration of the object (back object) and the target (front object). The image on the right shows the final configuration of the object with respect to its target.	26
2.19	The experimental setup for LDO manipulation used in [ZNF ⁺ 18]. In this work, a Fourier series is used to parameterize the shape. The shape parameters are used to estimate a local deformation model of the cable online. A velocity control rule is applied to the robot using the deformation model to deform the cable into the target.	27
2.20	A schematic of an LDO and the used robot (Campero) in the current research.	27
3.1	The studied object in this dissertation. p_1 and p_2 represented the position of the LDO and p_3 shows its angle.	31
3.2	Schematic diagram of the proposed method.	32
3.3	The LDO is discretized into a set of nodes (n_g gripped nodes, n_s servoed nodes, and n_f free nodes). The nodes are grouped in sets of three, which we call each of these sets a triad.	32
3.4	Geometric modeling. Left: four nodes in two triads $\mathcal{T}_k, \mathcal{T}_{k+1}$, with their angles θ_k, θ_{k+1} . Center: a generic triad. Right: close-up of a region of the center plot. Nodes are shown as circles, gradient vectors as arrows.	37
3.5	The experimental setup used in this chapter. Control point(s) and the grasped point are tracked with a camera.	38
3.6	Simulation 3.1. Pose control of one point. The object is modeled using ARAP ($L = 0.41(m)$). From the top-left, the initial and final shapes, the errors, and the gripper velocities on different axes are shown. The controlled point can be found as a red circle with a dashed line segment used to show its angle. The target point can be found as a blue diamond with a dashed line segment used to show the target angle.	39
3.7	Simulation 3.2. Pose control of one point. The object is modeled using ARAP ($L = 1.00(m)$). The initial and final shapes can be seen in the first row. The controlled point can be found as a red circle with a dashed line segment used to show its angle. The target point can be found as a blue diamond with a dashed line segment used to show the target angle. The errors are plotted in the second row.	39

3.8	Simulation 3.3. Pose control of one point. The object is modeled using KER ($L = 0.90(m)$). The initial and final shapes can be seen in the first row. The target point can be found as a blue diamond with a dashed line segment used to show the target angle. The errors are plotted in the second row. . .	40
3.9	Experiment 3.1. Pose control of one point. From the top-left, the initial and final shapes, the errors, and the gripper velocities on different axes are shown. The target point can be found as a red circle with a dashed line segment used to show the target angle.	40
3.10	Experiment 3.2. Pose control of one point. The initial and final shapes can be found in the first row. The errors are plotted in the second row.	41
3.11	Experiment 3.3. Pose control of one point. The initial and final shapes can be found in the first row. The errors are plotted in the second row.	41
3.12	Simulation 3.4. Position control of two points using two grippers. The object is modeled using KER ($L = 0.70(m)$). The grippers are shown as red and green circles having a dashed line representing their angles. From the top-left, the initial and final shapes, the error of the controlled points, and the gripper velocities on different axes can be seen. In the top-left image, the target points can be seen as a red diamond (as the target for the lower controlled point (shown in light blue)) and a blue diamond (as the target for the upper controlled point (shown in brown)). Subscript 1 is used for the lower controlled point with the red target and the lower gripper, and subscript 2 is used for the upper controlled point with the blue target and upper gripper.	42
3.13	Simulation 3.5. Position control of two points using two grippers. The object is modeled using KER ($L = 0.70(m)$). From the top-left, the initial and final shapes, the error of the controlled points, and the gripper velocities on different axes can be seen. In the top-left image, the target points can be seen as a red diamond (as the target for the lower controlled point (shown in light blue)) and a blue diamond (as the target for the upper controlled point (shown in brown)). Subscript 1 is used for the lower controlled point with the red target and the lower gripper, and subscript 2 is used for the upper controlled point with the blue target and upper gripper.	43
3.14	Simulation 3.6. Position control of two points using a gripper. The object is modeled using ARAP ($L = 1.00(m)$). From the left, the initial shape, final shape, and the errors of the three controlled points can be seen. The target points can be found as red (for the first controlled point which is in the lower section of the LDO) and blue (for the second controlled point which is in the upper section of the LDO) diamonds.	43
3.15	Simulation 3.7. Position control of three points using a gripper. The object is modeled using ARAP ($L = 0.85(m)$). From the left, the initial shape, final shape, and the errors of the three controlled points can be seen. The target point can be found as red (for the first control point that is lower), blue (for the second control point), and green (for the third control point that is upper) diamonds.	43
3.16	Experiment 3.4. Position control of two points. From the left, the initial shape, final shape of the object, and the errors of the three controlled points are indicated.	44
3.17	Experiment 3.5. Position control of two points. From the left, the initial shape, final shape of the object, and the errors of the three controlled points are indicated.	44

3.18	Experiment 3.6. Position control of three points. From the left, the initial shape, final shape of the object, and the errors of the three controlled points are indicated.	44
3.19	Experiment 3.7. Position control of three points. From the left, the initial shape, final shape of the object, and the errors of the three controlled points are indicated.	44
4.1	Considered object in the current study.	49
4.2	Schematic diagram of the proposed adaptive control strategy.	51
4.3	Simulation 4.1. From the top-left in the first row, the initial and final shape of the object and the servoing error of the controlled point are illustrated. The desired state of the point m is shown as a purple diamond (position) and a dashed line segment (orientation). L is $0.70(m)$. Second row shows the states of the controlled point regarding their desired trajectories in the first simulation using the KER model.	54
4.4	Simulation 4.2. From the top-left in the first row, the initial and final shape of the object and the servoing error of the controlled point are illustrated. The desired state of the point m is shown as a purple diamond (position) and a dashed line segment (orientation). L is $0.90(m)$. Second row shows the states of the controlled point regarding their desired trajectories in the second simulation using the KER model.	54
4.5	Simulation 4.3. From the top-left in the first row, the initial and final shape of the object and the servoing error of the controlled point are illustrated. The desired state of the point m is shown as a purple diamond (position) and a dashed line segment (orientation). L is $0.90(m)$. Second row shows the states of the controlled point regarding their desired trajectories in the third simulation using the KER model.	55
4.6	Simulation 4. From the top-left in the first row, the initial and final shape of the object and the servoing error of the controlled point are displayed. The desired state of the point m is shown as a blue diamond (position) and a dashed line segment (orientation). $L = 1.00(m)$. Second row depicts the states of the controlled point based on the desired trajectories in the fourth simulation using the ARAP model.	55
4.7	Simulation 4.5. From the top-left in the first row, the initial and final shape of the object and the servoing error of the controlled point are displayed. The desired state of the point m is shown as a blue diamond (position) and a dashed line segment (orientation). $L = 0.43(m)$. Second row depicts the states of the controlled point based on the desired trajectories in the fifth simulation using the ARAP model.	56
4.8	Simulation 4.6. From the top-left in the first row, the initial and final shape of the object and the servoing error of the controlled point are displayed. The desired state of the point m is shown as a blue diamond (position) and a dashed line segment (orientation). $L = 0.43(m)$. Second row depicts the states of the controlled point based on the desired trajectories in the sixth simulation using the ARAP model.	56
4.9	The experimental setup used in this work.	57
4.10	Experiment 4.1. From the top-left in the first row, the initial and final shape of the object and the servoing error of the controlled point are illustrated. The desired state of m is shown as a red circle with a small line segment. Second row presents the evolution of the states in the first experiment with respect to their desired trajectories.	58

4.11	Experiment 4.2. From the top-left in the first row, the initial and final shape of the object and the servoing error of the controlled point are illustrated. The desired state of m is shown as a red circle with a small line segment. Second row presents the evolution of the states in the second experiment with respect to their desired trajectories.	58
4.12	Experiment 4.3. From the top-left in the first row, the initial and final shape of the object and the servoing error of the controlled point are illustrated. The desired state of m is shown as a red circle with a small line segment. Second row presents the states of the controlled point in the third experiment with respect to their desired trajectories.	59
4.13	Experiment 4.4. From the top-left in the first row, the initial and final shape of the object and the servoing error of the controlled point are illustrated. The desired state of m is shown as a red circle with a small line segment. Second row presents the states of the controlled point in the fourth experiment with respect to their desired trajectories.	59
4.14	Experiment 4.5. From the top-left in the first row, the initial and final shape of the object and the servoing error of the controlled point are illustrated. The desired state of m is shown as a red circle with a small line segment. Object's length (L) is $0.72(m)$. Second row presents the evolution of the states in the fifth experiment with respect to their desired trajectories. . . .	60
4.15	Experiment 4.6. From the top-left in the first row, the initial and final shape of the object and the servoing error of the controlled point are illustrated. The desired state of m is shown as a red circle with a small line segment. Object's length (L) is $0.99(m)$. Second row presents the evolution of the states in the last experiment with respect to their desired trajectories. . . .	60
4.16	Experiment 4.7. From the top-left in the first row, the initial and final shape of the object and the servoing error of the controlled point are illustrated. The desired state of m is shown as a red circle with a small line segment. Object's length (L) is $0.86(m)$. Second row presents the evolution of the states in the last experiment with respect to their desired trajectories. . . .	61
5.1	Desired shape prediction based on the necessitated motion.	64
5.2	Schematic diagram of the proposed control framework.	65
5.3	The shape of the studied object (shown as a green rod), and the final pose of the grasped point (shown as a violet circle with a small dashed line representing its angle). The grasped point is demonstrated as a red circle with a small black dashed line.	65
5.4	Simulation 5.1. As shown in the top-left plot, the target pose of the grasped point is set to $[-0.0741, 0.8923, 2.0069]^T$. The object's initial, target, and final shapes can be observed in the first row. The servoing error and the cost functions of the controller (E) are plotted in the second row.	68
5.5	Simulation 5.2. As shown in the top-left plot, the target pose of the grasped point is set to $[-0.3941, 0.7890, 2.6384]^T$. The object's initial, target, and final shapes can be observed in the first row. The servoing error and the cost functions of the controller (E) are plotted in the second row.	69
5.6	Simulation 5.3. As shown in the top-left plot, the target pose of the grasped point is set to $[0.2861, 0.8030, 0.8249]^T$. The object's initial, target, and final shapes can be observed in the first row. The servoing error and the cost functions of the controller (E) are plotted in the second row.	69
5.7	The experimental setup used in this work.	70

5.8	Experiment 5.1. The target pose of the grasped point is set to $[0.0466, 0.3691, 1.2392]^T$. The object's initial, target, and final shapes can be observed in the first row. The servoing error and the cost functions of the controller (E) are plotted in the second row.	70
5.9	Experiment 5.2. The target pose of the grasped point is set to $[-0.0383, 0.8395, 1.2654]^T$. The object's initial, target, and final shapes can be seen in the first row. The servoing error and the cost functions of the controller (E) are plotted in the second row.	71
5.10	Experiment 5.3. The target pose of the grasped point is set to $[0.3262, 0.7232, 0.5585]^T$. The object's initial, target, and final shapes can be seen in the first row. The servoing error and the cost functions of the controller (E) are plotted in the second row.	72
6.1	Approximating the DO, inspired by [TKK01].	76
6.2	Representation of the considered LDO in the current study.	77
6.3	The initial (I_s) and target (F_s) shape of the studied object.	77
6.4	Schematic diagram of the proposed control strategy.	80
6.5	Different used objects in the performed experiments: a foam rod ($L = 0.87$ (m)), three branches of different plants ($L = 0.40, 0.60, 0.38$ (m))	82
6.6	The experimental setup used in this work.	83
6.7	Experiment 6.1. The used object is a deformable foam rod. From the top-left in the first row, the initial shape, final shape, and servoing error can be seen. The target shape is shown in red. The evolution of the shape (average of its states p_{1ml} , p_{2ml} and p_{3ml}) with respect to its desired trajectory can be found in the second row.	84
6.8	Experiment 6.2. The used object is a deformable foam rod. From the top-left in the first row, the initial shape, final shape, and servoing error can be seen. The target shape is shown in red. The evolution of the shape (average of its states p_{1ml} , p_{2ml} and p_{3ml}) with respect to its desired trajectory can be found in the second row.	84
6.9	Experiment 6.3. The used object is a branch of a plant. From the top-left in the first row, the initial shape, final shape, and servoing error are displayed. The target shape is shown in red. The evolution of the shape (average of its states p_{1ml} , p_{2ml} and p_{3ml}) with respect to its desired trajectory can be found in the second row.	85
6.10	Experiment 6.4. The used object is a branch of a tree. From the top-left in the first row, the initial shape, final shape, and servoing error are displayed. The target shape is shown in red. The evolution of the shape (average of its states p_{1ml} , p_{2ml} and p_{3ml}) with respect to its desired trajectory can be seen in the second row.	86
6.11	Experiment 6.5. The used object is a branch of a tree. The initial shape, final shape, and servoing error can be seen in the first row. The target shape is shown in red. The controller inputs (u_1 to u_6) are illustrated in the second row. The evolution of the shape (average of its states p_{1ml} , p_{2ml} and p_{3ml}) with respect to its desired trajectory can be found in the third row.	86
6.12	Experiment 6.6. The used object is a branch of a plant. The initial shape, final shape, and servoing error are indicated in the first row. The target shape is shown in red. The controller inputs (u_i) are presented in the second row. The evolution of the shape (average of its states p_{1ml} , p_{2ml} and p_{3ml}) with respect to its desired trajectory can be found in the third row.	87

A.1	Simulation 1. Comparison of the proposed methods in chapters 3 (ASAP) and 4 (adaptive). The length of the simulated object (L) is $1.25(m)$. From the top-left, the initial shape, the error of two methods, the final shapes of two methods, and the evolution of the states can be seen. The target point is shown in purple.	109
A.2	Simulation 2. Comparison of the proposed methods in chapters 3 and 4. The length of the simulated object (L) is $0.60(m)$. From the top-left, the initial shape, the error of two methods, the final shapes of two methods, and the evolution of the states can be seen. The target point is shown in purple.	110
A.3	Simulation 2. Some of the system parameters' estimations \hat{a}_{kj}	110
A.4	Simulation 3. Comparison of the proposed methods in chapters 5 (optimal) and 6 (adaptive). $L = 1.08(m)$. The first row shows the initial and final shapes of the object. The second row demonstrates the evolution of the servoing error in these tests. In the third row, the object's deformation with respect to its desired shape is shown for the adaptive method.	111
A.5	Simulation 4. Comparison of the proposed methods in chapters 5 and 6. $L = 1.08(m)$. The first row shows the initial and final shapes of the object. The second row demonstrates the evolution of the servoing error in these tests. In the third row, the object's deformation with respect to its desired shape is shown for the adaptive method.	112
A.6	Simulation 5. Comparison of the proposed methods in chapters 5 and 6. $L = 1.08(m)$. The first row shows the initial and final shapes of the object. The second row demonstrates the evolution of the servoing error in these tests. In the third row, the object's deformation with respect to its desired shape is shown for the adaptive method.	112
A.7	Simulation 5. Some of the system parameters' estimations \hat{a}_{kj}	113

List of Tables

3.1	State of the controlled points in the performed tests for controlling one point.	41
3.2	State of the controlled points in the performed tests for controlling multiple points.	45
4.1	State of the controlled points in the performed tests.	56
4.2	State of the controlled points in the performed tests.	61
5.1	Final error of the performed tests.	71
6.1	Variables used in the proposed controller scheme	78
6.2	Initial value, target value, and final error of $p_{1s}(m)$, $p_{2s}(m)$, and $p_{3s}(m)$ in the performed tests.	87
A.1	State of the controlled points in the performed simulations.	111
A.2	Final error of the performed tests.	113

Glossary

2-D two-dimensional. [4](#), [14](#), [17](#), [22](#), [25](#), [26](#), [29–33](#), [46](#), [49](#), [63](#), [73](#), [75](#), [89–91](#), [111](#)

3-D three-dimensional. [6](#), [11](#), [21](#), [22](#), [24](#), [25](#), [28](#), [33](#), [73](#), [75–77](#), [83](#), [85](#), [87](#), [90](#), [91](#)

ARAP As-Rigid-As-Possible. [xv–xviii](#), [6](#), [15](#), [16](#), [18](#), [19](#), [21](#), [32](#), [33](#), [35–37](#), [39](#), [43](#), [49](#), [55](#), [56](#), [62](#), [67](#), [68](#), [90](#), [91](#), [111](#)

ASAP As-Similar-As-Possible. [xxi](#), [32–37](#), [89](#), [109](#), [110](#)

BEM boundary element method. [xv](#), [12](#), [14](#), [15](#)

DO deformable object. [v](#), [vii](#), [xv](#), [xx](#), [1–4](#), [9–15](#), [19–24](#), [27](#), [30](#), [32](#), [35](#), [46–48](#), [55](#), [63](#), [64](#), [73](#), [75](#), [76](#), [79](#), [83](#), [89–91](#)

DOF degrees of freedom. [v](#), [vii](#), [2](#), [14](#), [23](#), [28](#), [34](#), [42](#), [67](#), [78](#), [91](#)

FEM finite element method. [xv](#), [11–15](#), [19–21](#), [23](#), [33](#), [35](#), [75](#)

FPS frames per second. [68](#), [69](#), [109](#), [111](#)

KER Kirchhoff elastic rod. [xvii](#), [xviii](#), [15](#), [19](#), [37](#), [40](#), [42](#), [43](#), [49](#), [53–55](#), [62](#), [109](#)

LDO linear deformable object. [v](#), [vii](#), [xv–xvii](#), [xx](#), [2–6](#), [9](#), [10](#), [15–20](#), [23–33](#), [38](#), [43](#), [46–48](#), [50](#), [57](#), [62–67](#), [70](#), [72](#), [75–77](#), [79](#), [87](#), [89–91](#)

MSD mass-spring-damper. [xv](#), [12](#), [13](#), [19](#), [20](#)

MSE mean squared error. [42](#), [45](#), [59](#), [61](#), [62](#), [71](#), [85](#)

Chapter 1

Introduction

The work accomplished in this dissertation is funded by a public grant overseen by the French National Research Agency as part of the "Investissements d'Avenir" through the IMobS3 Laboratory of Excellence (ANR-10-LABX-0016) and the IDEX-ISITE initiative CAP 20-25 (ANR-16-IDEX-0001). It is developed within the context of the project "SyncEA" ("Dynamic SYNChronization of mobile manipulators for new Environmental and Agricultural tools") of CAP 20-25 Challenge 2 (Domain: Agro Technologies). In the agricultural context, the ambition of the project is to participate in laying the foundations of new modes of agricultural production. It is also aimed to be beneficial in other similar application fields (e.g., factory of the future). It is part of the momentum of the dynamics launched locally on mobile handling and benefits from equipment already acquired under the CPER 2015-2020, MMII project (Mobility in Industrial and Uncertain Environments), Challenge MMAsyF (Mobilities - Materials and Systems of the Future).

The context of the SyncEA project concerns agro-technologies. It aims to open future perspectives for developing fast, precise, and safe tasks performed by autonomous mobile manipulators moving on agricultural land. In fact, the agro-ecological transition is a significant challenge for our society, and its implementation requires new tools to change production methods towards new, more ecological, and responsible methods. In this context, the contribution of robotics appears as essential support, and robots are needed in terms of multitasking movement and the ability to manipulate natural elements. Thus, the work focuses on two complementary aspects, and as a result, INRAE (UR TSCF) and Institut Pascal offered two robotics Ph.D. positions. The first one concerns the synchronization of the movements of a mobile manipulator for the precise and robust positioning of the effector environments (it is funded by the MMAsyF program, dealing with the navigation of mobile manipulators in relation to the SyncEA project). The second one, which is the topic of the current dissertation, deals with the manipulation of [deformable objects \(DO\)](#) for the realization of operations on plants. The main focus is to control one-dimensional [DOs](#) grasped by a mobile manipulator, with application in an agricultural domain.

1.1 Context

Robotic manipulation has come to the attention of researchers in recent decades since robots can undertake activities that are either risky or monotonous for humans. Until recent years, most works in this area have been focused on manipulating objects, assuming they are rigid (meaning that the shape of the object remains unchanged during manipulation), and researchers have suggested successful ways to meet the challenges [[BMAK14](#), [BK19](#)]. However, there is a wide range of objects whose deformation cannot be neglected during the manipulation process (such as clothes, human tissues, vegetables, metal sheets, cables, and so on). These types of objects have been named

deformable (as opposed to rigid) objects. Fig.1.1 shows several examples of humans manipulating DOs.



(a) Knot tying



(b) Fruits picking



(c) Clothes folding



(d) Cable deformation [CP20]

FIGURE 1.1: DOs manipulation examples

Automating DOs manipulation could significantly impact many human life applications as they are ubiquitous. Examples of DO manipulation range from agricultural tasks (for instance, in fruit harvesting) to medical tasks (such as robotic surgery). However, when it comes to robotic manipulation, effectively deforming these types of objects is very challenging. This is due to the fact that they have a high number of *degrees of freedom* (DOF), they can go under self-occlusions, there are local deformations in each part of these objects, and finally, the types of deformations vary from one to another.

Generally, the main research topics of robotic manipulation of DOs can be categorized as modeling, sensing, and control. DO modeling is mainly a study area in computer graphics. The fundamental difficulty in this area is that a realistic model that includes physical features is frequently computationally costly and unsuitable for real-time simulation [MM07]. Sensing the objects' deformation is typically achieved using visual or force sensors, which is not always possible. Finally, controlling the deformation of the objects and manipulating them desirably is the most critical challenge. It is important to note that our specific topic within this dissertation is the manipulation of *linear deformable objects* (LDOs) especially with agriculture as the main target field. Therefore, the main focus of the current study is on the shape controlling of *linear deformable objects*; yet, we develop or adopt the necessitated models to use in our controller where needed, and we also deployed the required detection techniques to sense the object.

1.2 Motivation

Most agricultural elements (such as branches) can be assumed as LDOs and many agriculture tasks need to reshape objects [Dav06, HS18, ZGT+18]. In this field, skilled labor is scarce, and labor costs are increasing significantly. Meanwhile, worker safety is another issue in traditional farming. Therefore, mechanical or robotic solutions for reducing the amount of hand labor seem necessary [HS18]. The problem is that this topic (i.e., robotic manipulation of DOs in the agricultural field) has not received significant attention in the former studies, and none of the existing LDO manipulation approaches have considered agricultural applications in their studies [AWH+18b, SCBM18, ZNAPC21, NALRL13b, Ber13, LWL19].

In the literature, several successful methods can be found that attempt to manipulate DOs in various applications. However, most proposed approaches require some information about the object (an exact model or an estimation of the deformation). They mostly need to perceive its total shape during the deformation process. Therefore, applying these methods with these limitations to the objects that exist in the agriculture field is not easy. This inspired us to develop methods to solve these issues. Moreover, this dissertation aims to set a milestone in using these methods for the tasks that involve the manipulation of objects in agriculture, such as tree pruning. Pruning is the most prevalent method of tree maintenance. Pruning must be done carefully since incorrect pruning can cause long-term harm to the tree or limit its lifespan. The main purpose of this task is to bend the branches so that the cutting point moves to an appropriate cutting location. It is worth stressing that the proposed approaches can also be used to manipulate the stem or branches of a vegetable plant for plant inspection and fruit harvesting tasks. The main issues are that in these kinds of tasks, one cannot have the model or the required offline information of the object each time.



(a) Pruning trees



(b) Fruit harvesting



(c) Branch inspection



(d) Wire harness

FIGURE 1.2: Examples of manipulating LDOs

It is important to note that this type of object is one of the most common **DOs** in industrial applications as well. Therefore, taking a step forward in the manipulation of **LDOs** can be useful in other domains, and the presented methods can also be used in other similar fields. Fig.1.2 shows several examples of **LDO** manipulation in different applications.

To close this section, it is worth highlighting that although there exist several studies on **LDOs** manipulation, there exist many challenges to deal with in this area. In the agriculture field, there exists a wide range of **LDOs** with different characteristics, and in some cases, deformation of the entire shape is needed while being manipulated. On the other hand, tracking their entire shape in some other tasks is not trivial (and not necessary). Therefore, this dissertation attempts to propose methods that enable robots to do the deformation tasks by considering these challenges.

1.3 Contribution and outline

Fig. 1.3 displays a simplified illustration of the current thesis. It outlines the key points of each chapter of this thesis sequentially. Furthermore, the contributions of this thesis are summarized as follows.

The second chapter provides related works on **DO** manipulation. We also investigate the agriculture robotics and the existing studies in manipulation of **DO** in this field. This chapter also introduces the models that we adopt to simulate the **LDO** in the rest of this thesis for development and validation of the proposed algorithms.

In the third and fourth chapters of this dissertation, we focus on deforming **LDOs** toward the targets by sensing only the controlled points and without perceiving their entire shape.

In the third chapter, we first propose a method to calculate a deformation Jacobian for **LDOs**, and then we use a proportional controller to deform the object by moving one endpoint of the object. This chapter's primary novelty is an offline technique for computing the Jacobian based on the rest shape of the object. Furthermore, a simple controller is designed by inverting the Jacobian model. Although this method has some limitations, it can be used to design different algorithms that do not require to perceive the entire shape. Our main contribution to this chapter is:

- A novel method for **2-D** control of the shape of elastic linear objects is presented based on an offline geometric model which, at run-time, does not require computing a deformation model or sensing the full shape of the object.

In the fourth chapter, we propose an adaptive control strategy to regulate the state (position and orientation) of an arbitrary point on the body of the object. The proposed method does not need prior knowledge of any model parameters of the object and works in real-time. There are several contributions in this section:

- This method can track a desired dynamic evolution of the state while the existing adaptive approaches in the literature, simply reach a fixed set-point without considering any deformation trajectory.
- Using the present technique, the Jacobian is not required to determine the link between the object's deformation and the gripper's motion. The provided controller does not require offline information of object's deformation.
- The present approach can control the full states of the controlled point, including the angle (i.e., we control the angle of the controlled point as well as its position). It contains a formal analysis of the system dynamics under the controller.

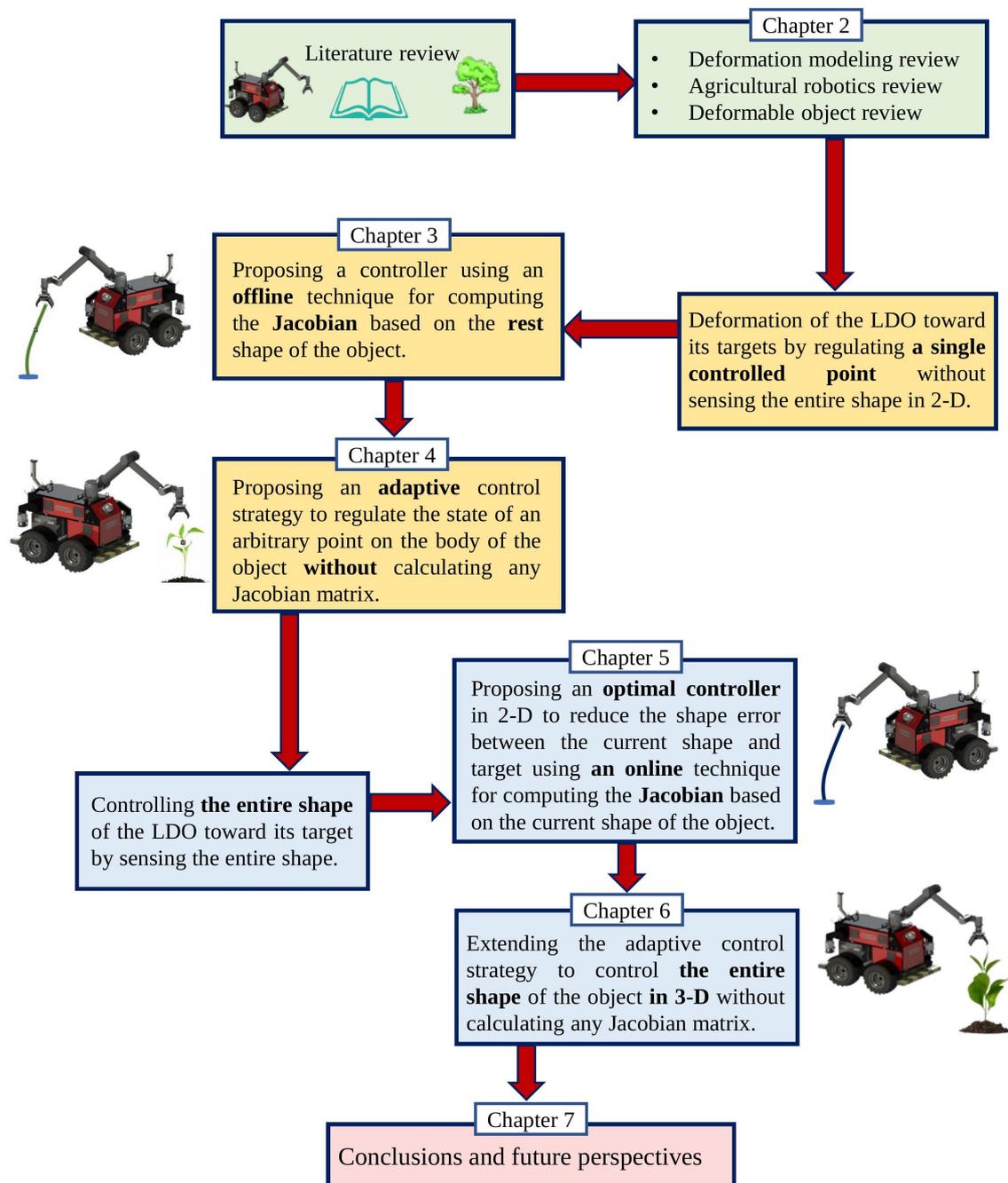


FIGURE 1.3: The schematic diagram of the present thesis that depicts the chapters in sequence.

The purpose of the fifth and sixth chapters of this dissertation is to go a step further than the previous chapters by focusing on controlling the entire shape of the object. Our goal is to provide methods that can be used to deform LDOs while the main focus is agriculture.

In the fifth chapter, we first propose a method to calculate a deformation Jacobian for LDOs, and then based on the proposed Jacobian, we develop an optimal controller to reach the target. In this chapter, we attempt to provide a complete methodology that can be used in the future to manipulate agricultural LDOs, such as branches and twigs of vegetables, without damaging them. The first section of this chapter is a shape prediction

optimal method that obtains a target shape that minimizes the stress along the target's length. Using this method, the reachability of the target shape can be guaranteed. The second part of this chapter is executed later and is based on an indirect optimal controller that automatically drives the objects' shapes into the target shapes by minimizing a cost function that reduces the error between the targets and the current shapes. To find the relation between the motion of the robotic arm and the object's shape, an online Jacobian matrix is calculated by using the *As-Rigid-As-Possible* (ARAP) [SA07] deformation model. This Jacobian does not have the limitation of the Jacobian of the third chapter and can be used with a broader range of objects. The contributions of this chapter are summarized as follows.

- The first part of this chapter attempts to address the problem of finding desired target shapes for the LDOs. We design a method to obtain a reachable target shape that is at minimum stress. This helps to have a stable shape for the target that is less likely to break.
- The main goal of the second part of this chapter is to propose an optimal control strategy to deform LDOs from initial shapes to the obtained target shapes. The method we use is to move mesh nodes on the object's body using the manipulated point to achieve the desired shape of the object. The proposed method does not need prior knowledge of any dynamical parameters of the object. Our method is simple to implement, and it does not necessitate collecting the data from a time window to compute a Jacobian.

In the sixth chapter, we extend the used adaptive scheme in chapter four to control the entire shape of the object in 3-D. Our objective is to propose a 3-D model-free method that can be used with a wide range of objects. Our contributions are:

- The proposed method can be used to track a desired dynamic evolution of the entire shape of the object in 3-D, rather than simply reach fixed set-points. This gives closer control on the path to be followed and on the time to complete the task.
- Our method works in real-time and is suitable for large deformations of the entire shape. Our proposed method controls the full shape of the object and not a simplified representation of it.
- The method we present does not require offline information about the object's deformation. In addition, we do not need to compute any Jacobian matrix to know how displacements of the manipulator are mapped to deformations of the body.
- The proposed method is suitable for manipulating objects that exist in agricultural domains, such as stems or twigs.

Upon review of all chapters, the last chapter of this dissertation summarizes the work developed throughout the current research and gives perspectives on future studies.

1.4 Publications

- Adaptive deformation control for elastic linear objects.
Omid Aghajanzadeh, Miguel Aranda, Juan Antonio Corrales Ramon, Christophe Cariou, Roland Lenain, and Youcef Mezouar.
Frontiers in Robotics and AI 9:868459.2022. doi: 10.3389/frobt.2022.868459
Video: <https://www.youtube.com/watch?v=mMsRmehdcXo> [AACR⁺22]

- Optimal deformation control framework for elastic linear objects.
Omid Aghajanzadeh, Guillaume Picard, Juan Antonio Corrales Ramon, Christophe Cariou, Roland Lenain, and Youcef Mezouar.
2022 IEEE 18th International Conference on Automation Science and Engineering (CASE), IEEE, 2022, pp. 722–728
Video: <https://www.youtube.com/watch?v=a3XoJFYlyc4> [APR⁺22]
- An offline geometric model for controlling the shape of elastic linear objects.
Omid Aghajanzadeh, Miguel Aranda, Gonzalo Lopez-Nicolas, Roland Lenain, and Youcef Mezouar.
Accepted by IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 2022.
Video: <https://www.youtube.com/watch?v=LI8JdTFPIR8> [AALN⁺22]
- 3-D shape control of linear deformable objects using an adaptive Lyapunov-based scheme.
Omid Aghajanzadeh, Mohammadreza Shetab-Bushehri, Miguel Aranda, Juan Antonio Corrales Ramon, Christophe Cariou, Roland Lenain, and Youcef Mezouar.
Submitted to Control Engineering Practice journal. 2022.
Video: https://www.youtube.com/watch?v=FIEMAy_IcZo

Chapter 2

State of the art and preliminaries

This chapter provides an overview of relevant methodologies to this topic. We discuss robotics in agriculture in section 2.1, despite the fact that DOs topic is new in agriculture, and there are few works on agricultural manipulation of DOs (mostly on picking application). Next, we review the works on DO modeling in section 2.2. At the end of this section, the models we use to simulate the objects in this work are described. Section 2.3 explores object manipulation deformation techniques generally addressed in other applications. Subsequently, the works on linear deformable object (LDO) manipulation, which is the main topic of this research, are explicitly presented in section 2.4. Finally, the work accomplished during this Ph.D. is positioned concerning the existing literature in section 2.5.

2.1 Robotics in agriculture

Agriculture is the backbone of many countries, which helps them to improve their economic and social situation. Agriculture is also one of the most important reasons for bringing people together, which has led to the creation and development of human civilizations worldwide over the past 10,000 years. The modern agricultural industry that the world has today with advanced, accurate, and quality technology results from changes in time and various inventions in agriculture. Today, environmental agriculture with modern advanced technology produces good quality food and meets the basic nutritional needs of human health. These significant changes helped the agricultural sector grow rapidly by discovering incredible innovations and creating various revolutions around the world [Bac15]. However, decreasing farmer and agricultural labor populations due to various factors is a severe problem in the present era. Skilled labor is scarce in this field, and labor costs increase significantly. Meanwhile, worker safety is another issue in traditional farming. To solve this problem, researchers aim to use highly advanced robots that can replace manpower to provide long-term solutions to agricultural mechanization and automation. Therefore mechanical or robotic solutions for reducing the amount of hand labor seem necessary [HS18]. It is notable that most agricultural activities can be performed by robots [SSM21].

The concept of using robotic technology in agriculture is relatively new. There are various applications of using robots in agriculture. Therefore, robots appear in various forms and increasing numbers in this field. Farmers are interested in utilizing robots to harvest their crop, gather fruit, or even take care of their animals. Moreover, the employment of robots in the agricultural sector can result in higher-quality fresh food, reduced production costs, and less need for human labor. Other advantages of using robots in agriculture are as follows.

- The robots can effectively detect better-quality items.
- They can be used to automate manual operations that are too risky for the operators.

- They do not get ill, get tired, or need time off.
- They offer fewer mistakes and work at greater speeds.
- They use less energy and generate less waste in comparison to the bigger, inefficient existing machinery.
- They can prevent pollutants and emissions by using clean energy and streamlining the production process.

Agricultural robots are complex systems that require interdisciplinary collaborations between different research groups for effective task delivery in unstructured crops and plant environments [RSHP⁺18]. In spite of the fact that agriculture is one of the essential fields of humanity that provides food resources, it has not received enough attention from robotic researchers, and numerous obstacles remain. One of these impediments is the deformation of agricultural items, as many agriculture tasks need to reshape objects [Dav06, HS18, FSTB21, ARB⁺21]. By way of illustration, in recent years, researchers have tried to develop automatic harvesting robots for various fruits and crops [KYS⁺09, HBvT⁺14, HKL⁺12]. Still, they considered all existing objects as rigid objects and ignored the deformation of the objects in this vital field. This is a particularly intriguing topic because by considering the deformation of the existing items, it is feasible to limit the potential damage to trees in robotic activities while increasing the performed task performance. It should be pointed out that there are several reasons why automatic manipulation of agricultural DOs is complicated: existing methods cannot provide the dynamic properties of objects online, detecting the entire object is not a trivial task in all cases, and a wide range of objects with different physical properties exists.

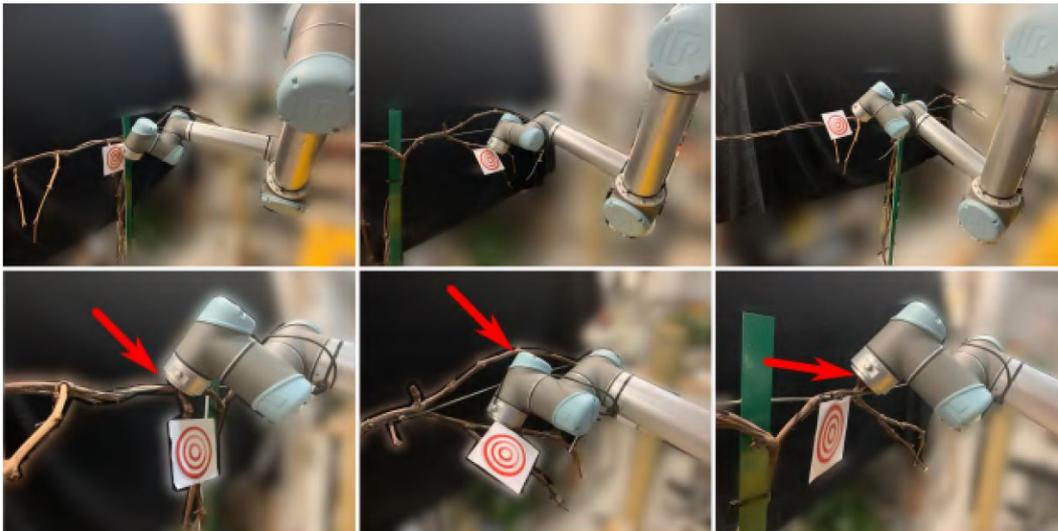


FIGURE 2.1: Several snapshots of the position of the robotic arm with respect to the target (red target board) presented in [YPS⁺21]. Red arrows show the contact points. The first row shows successful cases, and the second row shows failing ones where the end effector hits the tree.

In the agriculture sector, most agricultural elements (such as branches) can be assumed as LDOs. The problem is that there exists no work about the manipulation of this type of object (like the other types of agricultural DOs) in the former studies, and other existing LDO manipulation approaches [KFB⁺21, JWT19, SCBM18] did not consider this vital application in their robotic studies. Accordingly, this thesis aims to develop LDO manipulation approaches to be used in agricultural applications for some tasks such as pruning the trees. Pruning [YSF⁺20, BPG⁺17, YGSD22] is the most common tree maintenance procedure. Pruning must be done correctly because improper pruning can create lasting damage or shortening the tree's life. In this task, the main point is to deform the

branches so that the cutting point moves to where the branch can be cut appropriately. For instance, [BPG⁺17] described a robot system for automatically pruning grape vines without considering any deformation of the branches. Recently, [YPS⁺21] considered the pruning problem and developed an RL-based solution for the task of reaching goal points in vines by avoiding collisions with their branches. Fig. 2.1 presents various snapshots of some of the experiments done using this method for successful and failing cases where the goal was to reach a cutting point on the tree branches. Like the previous study, they did not take deformation of the objects into consideration.

Another application of our study can be in the robotic plant inspection and fruit harvesting tasks [BMBS16, BDB18, FZF⁺18]. Despite advances in agricultural robotics, millions of tons of fruits and vegetables are still hand-picked each year in open fields and greenhouses [SHKW18]. The problem with the current harvesting methods is that they cause some forms of damage, such as broken or removed stems that can affect the trees' life [ZWA⁺21]. [BDB⁺08] reported approximately 30% of the tree had stems removed during the harvesting of the apples. Moving the branches and picking the fruits without damaging the tree is necessary in this task. This means that branches and twigs should not be considered as rigid objects to be safely deformed. However, as mentioned, most agricultural robotic studies ignored this crucial fact.

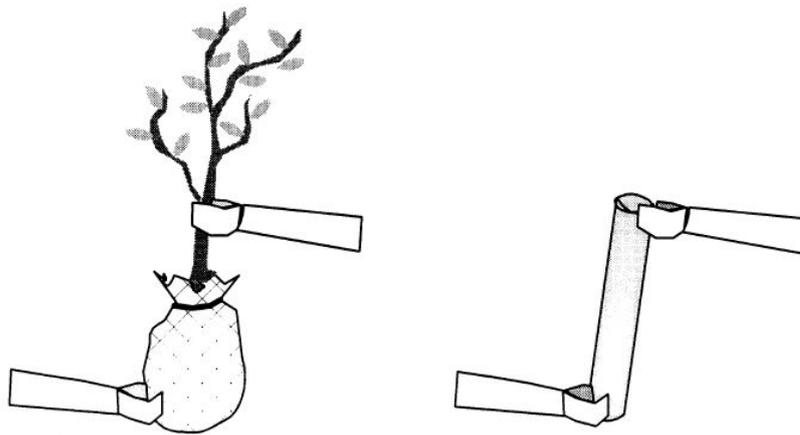


FIGURE 2.2: Approximating the DO as presented in [TKK01].

One of the first works in agriculture that assumed robots are dealing with DOs can be found in [TKK01]. The DO (shown in Fig. 2.2) is discretized using *finite element method* (FEM). However, that study mainly focused on obtaining equations of a multiple mobile manipulator system, and no shape controllers were presented. Another drawback of that work is that the model is not sufficient to guarantee nonholonomic motion for a real system, and additional work is needed to ensure the mechanical stability and nonholonomic motion of the robots. Recently, [SSM21] presented a system capable of generating 3-D models of non-rigid corn plants, which can be used as a tool in the phenotyping process. The method implemented to perform the model deformation was the Smoothed Rotation Enhanced As-Rigid-As Possible (SR-ARAP) [LG14] technique (like the previous work, no shape controller was provided). To the best of the authors' knowledge, no other relevant study considers the deformation of agricultural objects. As a result, in the next sections, we will go through the studies of DO manipulation that have been done in other applications in general. In the first step, we review the well-known methods used to model and simulate DOs (in section 2.2). Next, in section 2.3, we review the works on DO manipulation.

2.2 Deformation modeling

Modeling of DOs is a research topic in robotics that has a long history in many applications such as computer graphics, virtual surgery, cloth animation, agriculture, and industry. In fact, due to the highly-dynamic behavior of non-rigid objects, modeling the deformation of these objects is one of the main challenges in robotics [HLNAS19]. The major challenge of this research topic is that incorporating the object's physical properties is often computationally heavy and unsuitable for real-time simulation [MM07]. Until a few decades ago, most works were presented in geometric models for rigid bodies. However, in 1986, [SP86] proposed an initial free-form deformation method that deformed arbitrary objects by distorting space in which they were contained. In 1987, [TPBF87] proposed a class of elastically DOs for non-rigid curves, surfaces, and solids that incorporated the physical properties of the object directly in a graphical object which provided a way to represent the shape and motion of the DOs. It was the first time that a model of DOs responded to external forces and naturally interacted with other objects. Without a doubt, the presented works set a new milestone in the reliable modeling of various objects.

Since the problem of DO is complex, no single model can be suitable for all kinds of problems that exist in this field. Due to this fact, several works which rely on simulating the deformation of various objects can be found in the literature. The most well-known methods that are repeatedly found in the literature can be listed as follows:

- Geometric models [LL08, BMGA10]
- Mass-spring-damper (MSD) models [TFH99, LAAB14]
- Finite element method (FEM) [YVK21, LGWJ15]
- Boundary element method (BEM) [JP03].

Despite the existing studies, it is still challenging to obtain the exact models of DOs. Thus, researchers have attempted to work on DOs without an exact deformation model [HSP18, COC⁺20] over the last few years. These model-free studies use some estimation of the deformation behavior.

Since modeling of DO is not the main topic of the presented research, we give a general overview of the DO modeling and briefly present the models we use to simulate the studied objects at the end of this part.

2.2.1 Geometric models

Modeling and simulating DOs most often rely on geometric representations. The geometric models typically use a set of points or nodes to create curves or surfaces, as indicated in Fig. 2.3. They have been widely adopted in literature [MVDBF⁺12, ZGS12, CGLX17, YGL⁺18]. As the name implies, they do not consider the physics that govern the deformation. One of the most essential advantages of these models is that they do not consider the mechanical properties of the object. This can be useful for objects whose physical properties cannot be obtained easily. However, the accuracy of the models is compromised for fast calculations.

2.2.2 Mass-spring-damper models

One of the most common methods to model DOs used in different applications [TW90, BW98, JL97] is the MSD model. It was applied to the simulation of skin and muscle [CHP89], facial modeling in computer graphics [Wat87], creation of animals' animated locomotion [TT94], and cloth modeling [BHW94]. In general, these models include a set of nodes with a specific mass that are connected to each other by massless springs. To give a few examples, [DS11] used an MSD model to simulate a DO. Their model was used

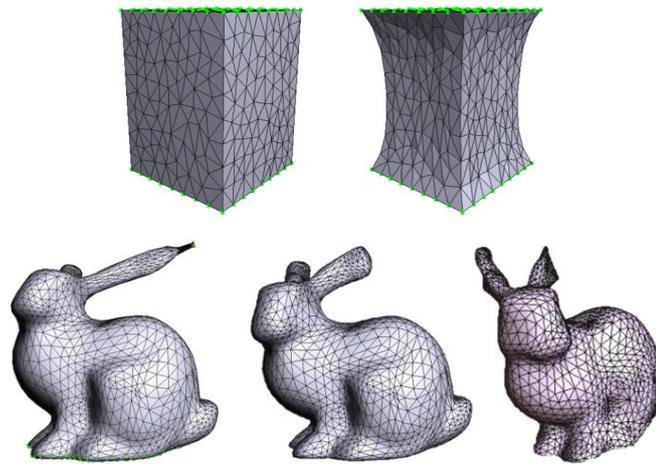


FIGURE 2.3: Different geometric models for elastic deformations presented in [YGL⁺18]. The objects' shapes were constructed using a set of geometrical nodes on their surfaces without considering any physical properties.

to deform a planar object toward a desired shape. However, they need a certain amount of information about the object to model it (see Fig. 2.4). Similarly, [HYK10] used an MSD model to investigate the deformation of a clay-like object.

One of the advantages of MSD models is that they can be easily implemented while being computationally efficient. However, the main drawback is that MSD models are not able to approximate the actual physics that occurs in a continuous body precisely. Another challenge is that this approach requires a significant amount of manipulation points (nodes to model objects' shape) to achieve acceptable results.

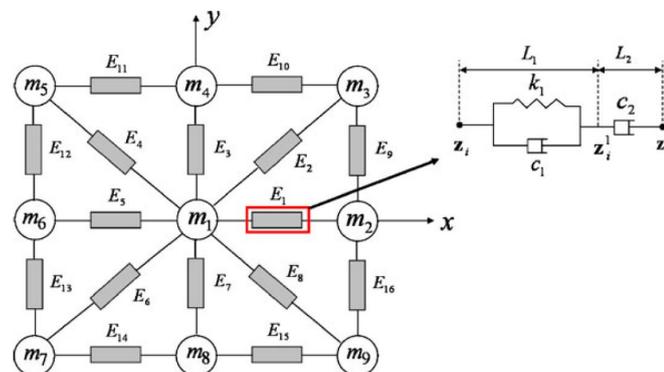


FIGURE 2.4: Model of an object with interconnected MSD presented in [DS11]. The mass points (m_i ($i = 1, 2, \dots$)) are inserted at the nodal points to model the object. Moreover, a three-element model (E_j ($j = 1, 2, \dots$)) is placed between the nearby mass points. E_j consist of elastic and viscosity coefficients of the non-residual deformation part (k_1 and c_1), and the viscosity coefficient in the residual deformation¹ part (c_2).

2.2.3 Finite element methods models

Another popular numerical method that has been widely used to model DOs is the FEM [FMC⁺18, SKM21, ZPC21]. It has a wide range of applications. To name a few, it has been utilized in skin simulation [BM17], muscle [CZ92], shape editing [CG91], cloth

¹An irreversible deformation that remains after the load is removed.

modeling [EKS03], and surgical simulation [BWG⁺99], which is an area where model accuracy is very critical. This method needs sets of physical properties (such as elasticity and physical properties of the object) to describe the object. In this method, a discrete mesh is used to divide the object into smaller, simpler parts called finite elements. The efficiency of the model is very sensitive to this discretization of the object. However, choosing the proper number of nodes is challenging because as the number of meshes increases, the model will be more accurate while the calculation time will also increase. Therefore, these models require significant tuning. Another point is that such models are dependent on the physical properties of the object, such as its Young's modulus and friction parameters. Fig. 2.5 shows an example of a flexible rod modeled using FEM in MATLAB.

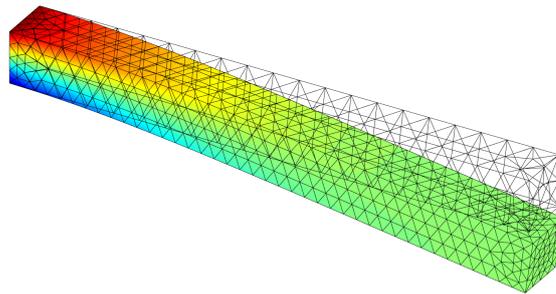


FIGURE 2.5: A flexible rod modeled in MATLAB using a large number of meshes to make the model accurate. However, due to the extensive use of mesh, it has a very high computational cost.

2.2.4 Boundary element method

In FEM, the objects have been modeled using the nodes positioned in their interior volume. However, in the cases where the DOF of the nodes on the surface (boundary) are needed, the BEM can be used to model the DO. By way of explanation, to mesh objects with BEM, only the boundary of the object is needed, while FEM requires the interior and boundary of the object simultaneously (as shown in Fig. 2.6). This property of BEM makes it suitable for computer vision problems because edge detection can be used to locate the boundary of an object. Another advantage of this method compared to FEM is that it can readily handle large deformations without mesh refinement [GSN04]. In this method, thanks to the Green-Gauss theorem, the three-dimensional problem can be reduced to two dimensions [Das10]. This leads to achieving a substantial speedup in the computational cost resulting in real-time simulation of global object deformation. However, BEM only works for objects that have a homogeneous interior. [JP99] utilized BEM approach to simulate volumetric models in real-time. Additionally, [GN03] developed a technique for 2-D tracking of DOs by using BEM to model deformation. Another application of this method can be found in [MMA⁺01], which has been used for surgical simulations.

2.2.5 Local models

Thus far, we have presented global models for DOs. These models are created using accurate offline data about the object. On the contrary, several recent works do not need an exact DO model by using some approximations of the deformation behavior. These local models are linear and can be computed in real-time with a small number of

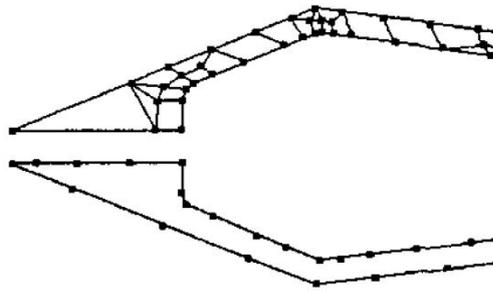


FIGURE 2.6: Comparison between BEM and FEM mesh presented in [GSN04]. FEM mesh is used to model the top jaw, and BEM mesh is used to model the bottom jaw.

data [ZCD⁺21]. They need to be continuously updated online during the object deformation. Some examples can be found in [MB20, ZNAPC21, COC⁺20]. Most local models approximate the perception/action relationship via a Jacobian matrix (a matrix that relates the motion of the robot with the changes in deformation of the grasped object). The main challenge is that since the model is assumed local, it should be continuously updated during task execution. Another challenge is to compute a Jacobian matrix that can be used for the entire process or continually updated during the deformation. The computation of an exact Jacobian is often computationally intractable and requires high-fidelity models. Therefore, its calculation should not be heavy in terms of computational cost so that one can use it in real-time controllers. Having these local models, one can design a simple controller by inverting the Jacobian model. However, since this controller only guarantees local convergence, the deformation cannot be so large.

Briefly, we have investigated the main advantages and disadvantages of the most well-known models used in the domain of DOs. We seek to model an elastic linear object by lowering the complexity. Based on this knowledge and reviewing the literature, we select two suitable techniques to model the studied object. We will merely adopt these models to simulate the object studied in this dissertation and evaluate our control strategies before performing the real experiments with the robots. These models will be succinctly presented in the section that follows.

2.2.6 Models used to simulate LDOs in this dissertation

In this dissertation, we aim to control the deformation of objects that can be thought of as LDOs (they are either linear, such as rods, or similar to linear objects, such as tree branches). Therefore, the objective of this section is to model an elastic linear object similar to what has been shown in Fig. 2.7.

To simulate the object, we consider a few assumptions. We presume that the ends' poses (position and orientation) of the object are known during the deformation. We assume that the object is symmetric in the simulation, while our control techniques can also be used with asymmetric objects. It is worthy to note that these assumptions are commonly used in the LDOs' manipulation literature such as [NAL13, NALRL13b, YZL22, LWL19]. We choose two fast and available models (Kirchhoff elastic rod (KER) and As-Rigid-As-Possible (ARAP)) among all the existing ones to simulate the LDO. KER [BM14, SS16] model is obtained based on boundary value methods which help us have a high computational speed in the simulations. On the other hand, ARAP [SA07] is a well-known model that has been used widely in computer graphics applications [SA07,

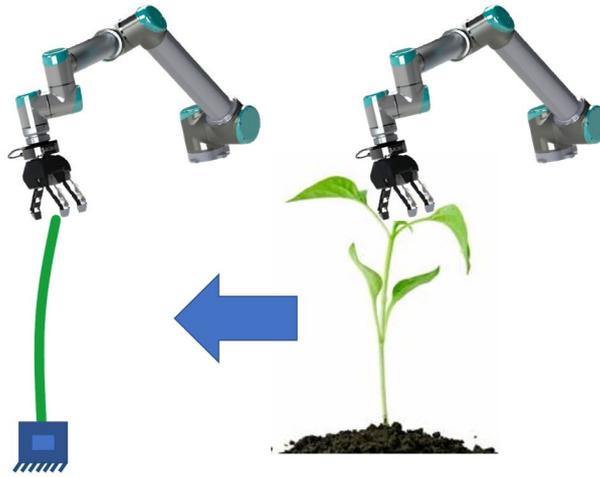


FIGURE 2.7: Approximating the studied object as LDO

[ACOL00, LG14]. The main idea of ARAP is to model the object geometrically while assuming the modeled object tends to preserve local rigidity. It is based on an energy measure that expresses the deviation from rigidity as the sum of deviations in local regions of the object [SBAMO22]. These two models can work in real-time. The reason we adopt two distinct models is to show that the proposed algorithms are not attached to any particular model. We use these models to validate our presented methods before applying them in real experiments. Therefore, our proposed controller methods in chapters 3, 4, 5, and 6 are independent of these models, as shown in Fig. 2.8. In the following sections, we will briefly introduce these models.

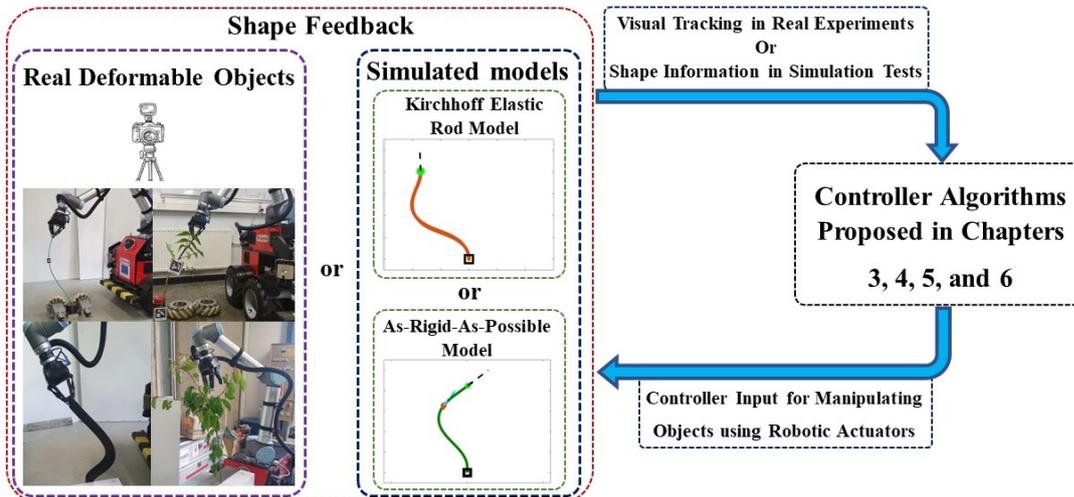


FIGURE 2.8: The schematic diagram shows the relationship between the proposed controller methods and the adopted simulated models/real objects in this thesis. As one can see, the controllers are designed independently, and the models or objects are used to validate the methods.

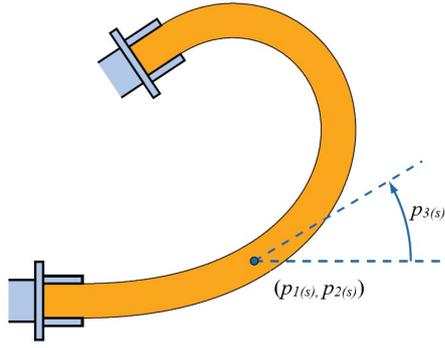


FIGURE 2.9: An elastic rod in static equilibrium, held at each end by robotic grippers presented in [BM14].

2.2.6.1 Kirchhoff elastic rod formalism

In this method, we assume that the object deforms like an elastic linear object. This method is commonly used to model objects of the considered type [BM14, SS16, WH04]. Hence, to model the LDO, we use a minimum-energy-based scheme to compute the shape of the object. According to [BM14, SS16], the shape of the LDO in 2-D could be defined by a curve as follows (see Fig. 2.9):

$$P(s) = [p_1(s), p_2(s), p_3(s)]^T \quad (2.1)$$

where p_1 and p_2 are the coordinates of the curve traced by the object at any point $s \in [0, L]$ (L is the length of the LDO), and p_3 is the tangent angle to this curve. The curve P must be a solution to the following curve optimization problem:

$$\min_{P,v} \int_0^1 (k \frac{v^2}{L^2} + LW_g p_2) ds$$

subject to

$$\begin{aligned} [dp_1/ds, dp_2/ds, dp_3/ds]^T &= [L\cos(p_3), L\sin(p_3), v]^T \\ P(0) = P_0 \ \& \ P(L) = P_L \in R^3 \end{aligned} \quad (2.2)$$

where k , W_g , and v are the bending stiffness, the weighting factor for gravity, and the curvature of LDO, respectively. By adopting the taken cost function, we try to minimize the total elastic energy over the whole body of the object. The problem of shape computation is identical to an optimization problem. Due to the maximum principle theory [BM14], there must exist a costate trajectory $f_i : [0, L] \rightarrow R^3$ (see [BM14] for more information), where $i = 1, 2, 3$. We set $[f_1, f_2, f_3]^T$ to represent the costates. By having this information and solving the given optimal problem (2.2), the formulation to obtain shape of the object can be found as [SS16]:

$$\begin{bmatrix} dp_1/ds \\ dp_2/ds \\ dp_3/ds \\ df_1/ds \\ df_2/ds \\ df_3/ds \end{bmatrix} = \begin{bmatrix} L\cos(p_3) \\ L\sin(p_3) \\ v \\ 0 \\ LW_g \\ Lf_1\sin(p_3) - Lf_2\cos(p_3) \end{bmatrix} \quad (2.3)$$

subject to the boundary conditions: P_0 and P_L .

(2.3) is adopted for LDO simulation with the aim of evaluating the efficiency of the methods presented in this dissertation.

2.2.6.2 As-Rigid-As-Possible

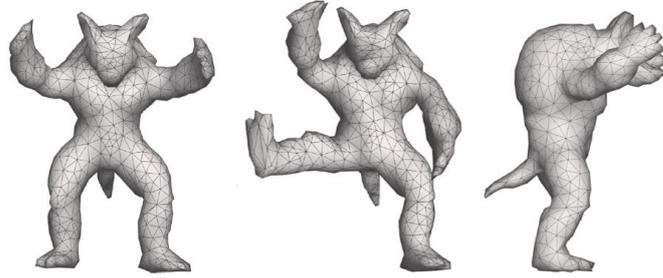


FIGURE 2.10: Modeling an Armadillo using ARAP presented in [SA07].

ARAP has been used repeatedly in diverse applications [LG14, PPBC15, DDF⁺17, HZSP18]. In this method, the object is modeled based on an energy measure (E) that expresses the deviation from rigidity as the sum of deviations in local regions of the object (so-called shell energy [TPBF87, SA07]). This model assumes that the object tends to preserve local rigidity even during the deformation. The main background idea of this model is to design a method according to the minimization of the shell energy.

To simulate the LDO based on ARAP, we use the algorithm presented in [SA07]. As mentioned, ARAP relies on E that expresses the deviation from rigidity. To calculate E , a cell is defined for each node in the mesh, comprising its first-order neighbors [SBAMO22]. This neighborhood is called \mathcal{N}_i . E_i , which expresses the deviation from the rigidity of the cell i , is defined here to show the energy of each cell in the object. E_i is:

$$E_i = \sum_{j \in \mathcal{N}_i} w_{ij} \| s_i - s_j - \mathcal{R}_i(s_i^u - s_j^u) \| \quad (2.4)$$

s_i shows the shape of node i of the modeled object. s_i^u is the deformed shape of s_i , which is assumed to be known before starting the task. \mathcal{R}_i stands for the effect of optimal rotation for every cell that minimizes E_i for the given information, and w_{ij} is a scalar encoding the connection between nodes i and j in the mesh [SBAMO22, SA07]. By having E_i for each cell, one can obtain E as follows:

$$E = \sum_{i=1}^n E_i \quad (2.5)$$

By minimizing the energy E_i for the given set of positions s_i , we reach to the following equation [SA07]:

$$\sum_{j \in \mathcal{N}_i} w_{ij}(s_i - s_j) = \sum_{j \in \mathcal{N}_i} \frac{w_{ij}}{2} (\mathcal{R}_i + \mathcal{R}_j)(s_i^u - s_j^u) \quad (2.6)$$

The linear combination on the left-hand side is non-other than the discrete Laplace-Beltrami operator [SA07] applied to s_i . The system of equations can be compactly presented as

$$Ls_i = b \quad (2.7)$$

where b is an n -vector whose i th row contains the right-hand side expression from (2.6). Having (2.7), one can simulate the object based on ARAP. Although most former works used (2.7) to model the volumetric objects with planar meshes (as in Fig. 2.10), in this work we use it to model an elastic linear object.

2.2.7 Conclusion

In this section, we have discussed the main benefits and drawbacks of the different methods to model the deformation of the objects in simulations. Among all the presented models, we choose two very different methods ([KER](#) model and [ARAP](#)) to use in our tests. The main advantages of these methods are that they are pretty fast in terms of computational cost, can precisely show the deformation of [LDOs](#), are capable of expressing a complete deformation, and have been successfully validated in the former studies. The main difference between these methods is that [ARAP](#) is faster than [KER](#) since it only uses the geometrical mesh to model the object. However, [KER](#) is more realistic since it contains some object properties. Furthermore, to model the object using [KER](#), we need to use more mesh than [ARAP](#), which is another reason for [KER](#) to be more computationally expensive than [ARAP](#). In the next section, the state of the art in manipulating [DOs](#) is reviewed.

2.3 State-of-the-art on deformable object manipulation

In this and the following sections, we review the existing literature on robotic manipulation of non-rigid objects. As a consequence, this section generally provides the works related to [DO](#) manipulation, whereas the next section particularly presents existing studies on [LDO](#) manipulation. Although shape control of a [DO](#) is a challenging problem that requires sophisticated algorithms, several successful methods of [DO](#) manipulation can be found in the literature. One of the first works in this field is [\[II84\]](#), which used visual feedback to propose a method for knotting problem. A few years later, [\[SNR96\]](#) proposed a relative elasticity model using the visual feedback to be used in [DO](#) manipulation without a physical model. At the beginning of the third millennium, a simple control law for the grasping manipulation of [DO](#) is presented in [\[HTW01\]](#) by utilizing multiple sensors (force and vision). A position/force hybrid control method that used visual information with force control is presented in [\[HTY05\]](#). [\[PML04, Cha04\]](#) employed an online estimation method to control the shape of the objects. However, the method of [\[PML04\]](#) is susceptible to entering local minima, and the real-time performance of [\[Cha04\]](#) is limited owing to a large number of calculations. Although these works tried to solve some of the existing primary challenges in this field by simplifying the problems, they created a clear path for future works.

In the last several years, many researchers have made efforts to design appropriate approaches to manipulate these objects [\[Tan06, ACM19, SMBB20\]](#). Some of the manipulation techniques available in the former studies have been proposed based on an exact deformation model. These works have been called model-based in the literature. As the name implies, these methods are based on a deformation model, assuming that the deformation properties must be known or estimated in advance during the calibration phase [\[SP09, P⁺11\]](#). These techniques are represented by stiffness or elasticity coefficients over different models such as [FEMs](#) and [MSDs](#).

Model-based controllers have been used for instance in [\[ZBD⁺17\]](#) for controlling soft robots. [\[LWL19\]](#) proposed a model-based method to reshape the object by moving some of its points toward their objectives. In this work, the multiple feature points are manipulated in a sequential manner, which means only one point is manipulated at each step, and the rest remain static. Using this technique and the Lyapunov theorem, the stability was proved. [\[DS11\]](#) employed an [MSD](#) model to reconstruct the shape of [DO](#). A robust controller is presented to minimize an energy-like criterion derived from the shape error between the desired and initial shapes. [\[KP12\]](#) used indirect [DO](#) manipulation to move a few points on the object to their targets. [\[JC07\]](#) adopted an [MSD](#) to model the object and

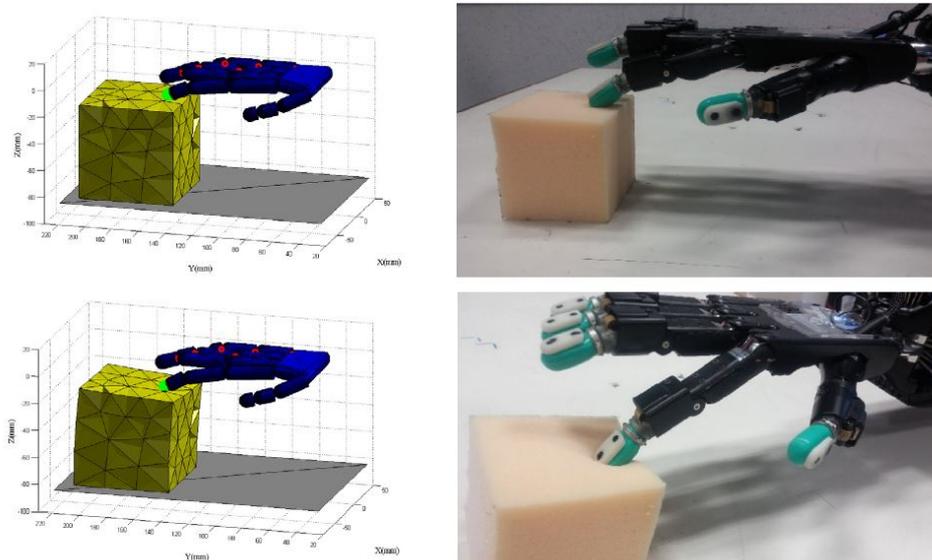


FIGURE 2.11: A deformable foam manipulation that was modeled with a non-linear mass-spring system [ZCB⁺17].

reported a locally optimal method to control the behavior of physically-based simulation of DOs. In [SH06], a symmetric linear MSD model was applied to analyze the dynamic behavior of the object in a simple way. Then, a PID controller was adopted to find an adequate feedback gain to stabilize the system. In [ZCB⁺17], the authors utilized a non-linear MSD model to represent the object deformations based on the mechanical loads generated by the robotic actuator. In this work, the initial task is to use MSD to determine the initial grasp configuration (as shown in Fig. 2.11), and then a position-control scheme was presented to move the fingers of the robotic hand towards their desired position.

An FEM simulation is used in [FSAB11] for effector path planning while preserving the shape of a DO. This work consists of offline and online steps. In the offline step, training data were generated as a preprocessing using the physical deformation simulation system, and a cost function was obtained that considers the distance between robot configurations and the deformation cost estimated using a FEM simulation tool. Subsequently, new deformation costs were calculated from learning ones using Gaussian Process regression during the online step. [SMEDC⁺20] proposed a pose control method for a DO using a force sensor and a physical model of the object. FEM physics-based simulation is also used in [AGDMC16] to insert a flexible needle automatically and slowly in deformable tissue using a predefined path.

[DBPC18] used an FEM model to represent an object (as seen in Fig. 2.12) and applied optimization to find the robot's joint angles that moved the object into the desired shape. This dual-arm manipulation method acts without relying on a feedback signal representing the object's shape. Similarly [FMC⁺18] used an FEM model to find a proper FEM-based deformation control for dexterous manipulation of DOs. Both of these works [DBPC18, FMC⁺18] executed open-loop control, and no feedback was used on the actual shape of the object. [LLJ22] proposed a modeling and control technique to control LDOs. First, a dynamic model of LDOs based on the discrete elastic rod model was proposed to simulate the stretching, bending, and twisting deformation of LDOs. Later, a dynamic control scheme for single-arm manipulation and dual-arm manipulation was developed to deploy the LDO onto the plane and form the target shapes at

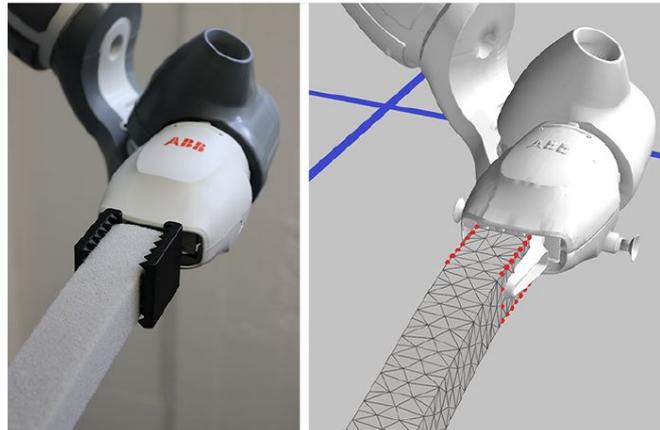


FIGURE 2.12: FEM model of the object used in [DBPC18] for a simulation-based control methodology. To model the object using FEM, some physical information about the object is needed in advance.

the same time. [ZPC21] proposed two trajectory optimization techniques based on Newton's method and DDP for robotic manipulation of DOs. The presented techniques are open-loop and depend on a forward simulation model of soft objects, which has been developed using a volumetric mesh based on the FEM method. Recently, [SBAMO22] proposed a method to control the 3-D deformation of DO on a deformation Jacobian computed from ARAP. In this study, a shape servoing method based on ARAP model was also presented.

Although several model-based methods exist in the literature that successfully manipulate DOs, the main drawback of these methods is that they need and are attached to prior knowledge about the system, including the object mass, geometry, or physical parameters. Another point is that these kinds of algorithms are usually expensive in terms of computation time. To avoid these obstacles, the methods that need minimum information about the objects can be used while providing fast calculations. Another solution is to use the model-free methods. Unlike the model-based methods, model-free techniques do not rely on any characterization of the object. These studies solve the need for an exact DO model by using some approximations of the deformation behavior [MB18, ZNF⁺18, COC⁺20]. These models are linear and can be computed in real-time with a small number of data [ZCD⁺21]. They need to be continuously updated during the object deformation since they are local.

Most of the studies in this domain used adaptive approaches or a Jacobian matrix that describes the relationship between robot motions and feedback deformations. Broyden rules [AWH⁺18a], receding horizon adaption [ZNAPC21], local gradient descent [NAL17], QP-based optimization [LKM20], and multi-armed Bandit-based methods [MB20] are some methods of model updating that existed in the former studies. To name a few, [NALRL13a] used parameter linearization and least-squares techniques to calculate the Jacobian deformation matrix. [HA94] used the Broyden method to online estimate the Jacobian matrix without a known structure. Kalman Filter is used in [LH06, WYW⁺18] to calculate the Jacobian with high performance in estimating unknown variables using a series of measurements observed over time in the presence of noise and uncertainty.

One of the most interesting successful model-free works to control the deformation of an object using a robot can be found in [NALRL14]. In this paper, two vision-based adaptive methods that do not need the prior identification of the object deformation model

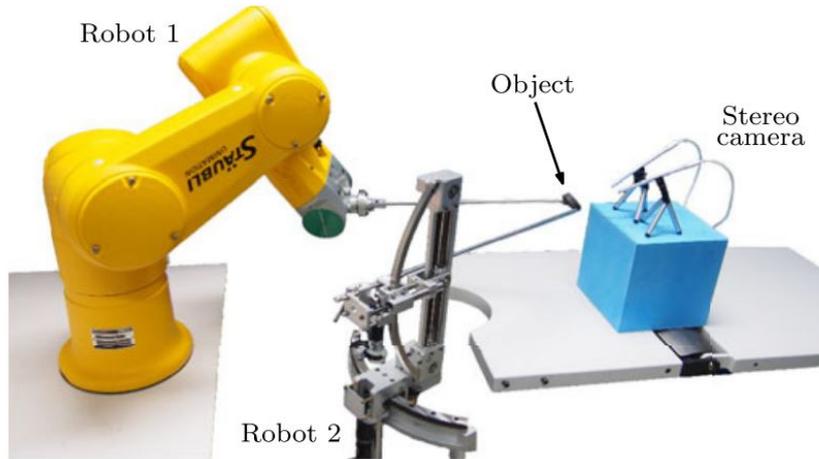


FIGURE 2.13: Setup used in [NAYW⁺16] for the automatic manipulation of the objects.

and the camera parameters were presented. The studied object assumed to be quasi-static and purely elastic. Although the controllers successfully performed the desired tasks, the main drawback of this work is that the presented method can only be used for small deformations. [NAL14] used an online estimation method to control the shape of the objects. Performance of this method necessitates a prior-known structure of the setup and depends on the choice of regression matrix. Later, an adaptive controller is presented in [NAYW⁺16] to deform an elastic object in 3-D, also without requiring the deformation model. The experimental setup of this work is illustrated in 2.13. In this article, based on the error between the deformation model and the velocity of the feature points, an estimation of the deformation model was computed, and a velocity controller was presented. One of the limitations of this method is that it is not effective for mono-camera experiments. Although the proposed method has taken a big step in manipulating DOs, it requires a proper choice of the feature points, and it is not able to handle large curvature deformations appropriately.

Later, [NAL17] proposed a vision-based controller to manipulate the entire shape of the object using its 2-D shape based on truncated Fourier series. This method can be used for cases where the shape of the object can be correctly described with a reduced number of Fourier coefficients. A robust data-driven control framework to indirectly manipulate heterogeneous 3-D compliant objects in the presence of unknown internal and external disturbances is presented in [AWH⁺18b]. Visual feedback is used to design the controller. This method does not need any prior knowledge of the object. The Jacobian matrix that relates the grasping points motions to the feature points motions is estimated online using the Broyden update rule. [FP04] developed a method based on a neural network to select appropriate grasping points on the DOs using visual feedback.

[DRCRM⁺17] described a model-free approach for DO manipulation according to tactile images. The proposed algorithm introduced a way to represent and use tactile information based on a combination of dynamic Gaussians that are defined from the sensor values. Model-free methods presented in [NP19, NP20] tried to move the 2-D contour of the studied objects toward the contour of the target. The main limitations of the presented model-free methods are that they are often very slow and cannot be used for large and complex deformations.

As we discussed some works that existed in the literature on DO manipulation, one can find out that most of the methods that allow a global control of the object deformation need physical models. Several works just control the 2-D shape of the object (based on a

visual projection of objects obtained by cameras) or require an offline step to be able to complete the task. In the next section, we investigate the literature of [linear deformable objects \(LDOs\)](#), which are more pertinent to the dissertation's subject.

2.4 LDO manipulation

A particular subdomain that has attracted considerable attention in the past few years is the manipulation of [linear deformable objects \(LDOs\)](#) such as cables, wires, flexible rods, plant stems [[LKM20](#), [QMZ+21](#), [KFB+21](#), [QMZ+21](#)]. This recent interest is due to the fact that many industrial and agricultural applications demand manipulating LDOs. To name a few, LDOs should be manipulated in assembling devices in 3C² manufacturing [[LSGL18](#)], surgical suturing [[CLP+20](#), [JÇ13](#)], automated handling of flexible materials [[SN02](#)], threading a needle [[WBB15](#)], knot tying [[SHLA16](#), [BLM04](#), [KW02](#)], and harnessing a cable [[TWT18](#)]. Therefore, in this section, we review the type of research that is closely related to the main topic of this dissertation which is LDO manipulation. Nevertheless, LDOs deformation toward desired shapes is very challenging for robots since the system is underactuated (typically, the number of controller inputs is less than the DOF of the controlled system). Hence, in cases where the objective is to control the entire shape or a part that includes several controlled features, the controllability of a closed-loop system is not guaranteed [[LWL19](#)]. Another tough challenge in this domain is to model these objects accurately and predict their deformation [[SCBM18](#)].

Different methodologies have been proposed to manipulate linear objects in the state of the art. For instance, several works about manipulation planning of LDOs can be found in [[LK01](#), [BM14](#), [BB15](#)]. In this regard, [[MK06](#)] developed a path planning technique for LDOs and explored the entire space of stable, collision-free configurations. [[SIL08](#)] offered a Probabilistic Roadmap [[KSLO96](#)] that plans for knot-tying tasks with an LDO. A cooperating dual-arm was used to tie self-knots and knots around simple static objects without assuming a specific physical model of the LDO.

Multiple popular and successful methods have been proposed based on a pre-existing deformation model to manipulate LDOs. [[PLK02](#)] is one of the first works in LDO manipulation, which presented a simple method of knot tying. In this article, a loosely knotted rope was pulled tight, and the knot was preserved, using an impulse model for collision handling. [[AZMD15](#)] presented a framework for aerial knot-tying using quadcopters, which is one of the fundamental tasks for the aerial assembly of tensile structures. Recently, [[KFB+21](#)] proposed a method using a reduced FEM to present a closed-loop shape control of LDOs. However, these methods depend on the analytical model's accuracy, and they need to accurately obtain LDOs' parameters. [[DHYH18](#)] assumed that LDO is a nonlinear system with a significant number of underactuated DOF and one actuated DOF. Given the LDO's motion characteristics, this paper stated that the vibration at the end of an LDO is unavoidable, and proposed a dynamic surface control strategy to control an LDO by trying to eliminate the vibrations. However, this algorithm is used for low-speed movements. An approach based on a diminishing rigidity assumption has also been proposed [[Ber13](#)], which attempted to move the object into a desired configuration using a pair of floating robot's grippers. Later, [[RMB18](#)] developed a more effective controller than previous work by constructing a more accurate geometric model of how the direction of gripper motion and obstacles affect DOs. The most crucial downside of these techniques is that they require a large amount of offline data in advance, which is

²Computer, Communication and Consumer electronics.

not always possible, particularly in the application we are working on. Moreover, at run-time, the cited methods require sensing the full shape of the object in order to simulate the deformation model.

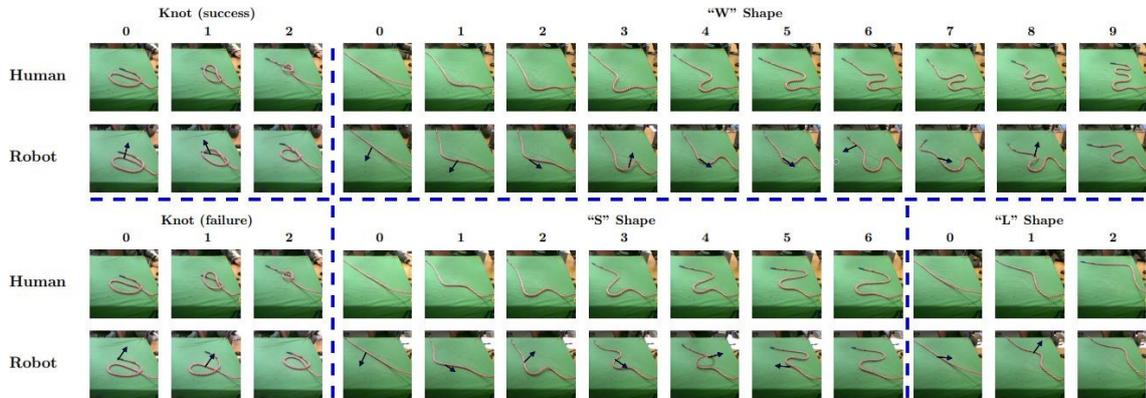


FIGURE 2.14: The robot’s performance in emulating the human demonstration for deforming the rope into various shapes was compared qualitatively and presented in [NCA⁺17]. The sequence of example images supplied as input to the robot is shown in the upper row. The lower row shows the states achieved by the robot as it attempts to follow the indicated trajectory.

Another prominent technique that has lately acquired attraction in the literature is addressing LDO manipulation problems via the use of machine learning [YZL22, LK21, HDL⁺21]. To mention a few examples, a deep-neural-network-based dynamics model was proposed in [YZJB20] to predict the next shape of the object. In [HHS⁺19], a multi-layer neural network encoded the mapping between end-effector motions and object deformations. [YZL22] proposed a scheme for LDO deformation control relying on both offline and online learning. The work [LK21] addressed LDO manipulation via the use of reinforcement learning and accommodated elastoplastic behaviors. A data-driven method to control a DO using learned nonlinear dynamics model is presented in [PMY⁺22]. [NCA⁺17] used a learning method according to the visual feedback to manipulate ropes. They have trained a predictive model of rope behavior using data of rope interaction collected autonomously by the robot, as shown in Fig. 2.14. This self-supervised model predicts which action the robot can execute to put the rope into the target configuration based on the current and target image of a rope. Some of the results obtained by this method are illustrated in Fig. 2.14, and they show that this method still has some limitations and is not able to achieve the target in all cases. The main drawbacks of the cited methods are that they are mostly complex, and applying these strategies to various objects is not straightforward. Training these approaches frequently necessitates a significant amount of time and effort. Finally, they mostly are not able to manipulate the objects into complex configurations.

The next studies that we investigate are the ones that tried to estimate the local linear deformation model of LDOs. They use some shape estimations to deform the objects [JWT19, LKM20, NAL17, ZNAPC21]. They mainly used some approximations of the deformation behavior. The problem is that only a small amount of local data can be used, so these estimated methods are less accurate. A feedback signal is required in these approaches to control the objects. Consequently, they mostly employ visual information as feedback since, in most applications, visual sensors are adequate to get the essential information and are inexpensive and non-invasive. Moreover, visual servoing techniques are very effective since the control loop can be closed over the vision sensor [TM10]. In this regard, [LKM20] proposed a method to deform a wire in 3-D using a least-squares

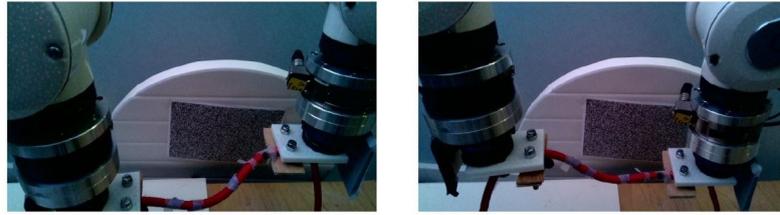


FIGURE 2.15: Experiment with large deformations of the wire presented in [LKM20]. The target shape is shown on the left, and the final shape of the wire is presented on the right. This figure depicts an example of unsuccessful attempts in large deformations with the approach provided in [LKM20].

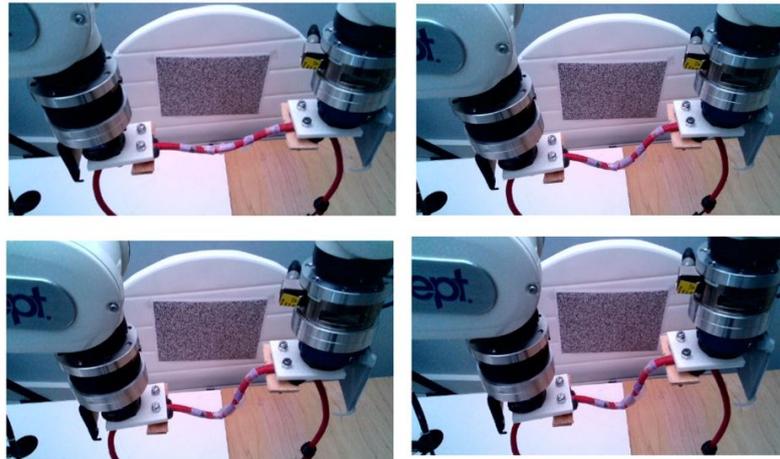


FIGURE 2.16: A successful test with intermediary targets for reaching large deformations of the wire that is presented in [LKM20]. Some of the intermediary targets are shown in the three first pictures. The final picture shows the final configuration of the wire that reached the target.

estimation to estimate the Jacobian matrix online. In the first step, this work proposed a visual feature that permits modeling the 3-D shape of a wire and based on the obtained information, a geometric model independent of the object's mechanical properties was used. Next, an adaptive model-free shape servoing method is presented to control the shape of a wire toward a desired shape. This method cannot directly control large deformations (as shown in Fig. 2.15). Intermediate targets are therefore used to achieve large wire deformations (as shown in Fig. 2.16).

[QMNA⁺20] proposed a vision-based algorithm to control the shape of elastic rods with robot manipulators. This study calculated parametric regression features from online sensor measurements and used an adaptive controller to estimate the transformation between the robot's motion and the relative shape changes. One of the experiments performed with this method is shown in Fig. 2.17. The main drawbacks of this method are that the algorithm is very slow, it is used only with small and simple deformations, it is sometimes not resistant to assumed noises and disturbances during deformation, and it only controls the shape of the object in 2-D without any gripper rotation.

[YZZL21] proposed a method for the manipulation of LDOs with unknown deformation models, in which the unknown deformation model is estimated using both offline and online learning approaches in order to combine the benefits of both. In another similar work, [YLZL22] has adopted a method for large deformation of LDOs with coupled offline and online learning of the unknown global deformation model. In the offline



FIGURE 2.17: An experiment performed with the approach presented in [QMNA⁺20]. The target is shown in red, and the green curves illustrate the path that the object took to get there from the initial shape.

phase, a model is trained with respect to different random movements of LDOs to obtain an estimation with acceptable generalization performance. Later in the online phase, the deformation controller was applied. The main limitations of this work are its slow manipulation, and the path of movement is not optimal globally.

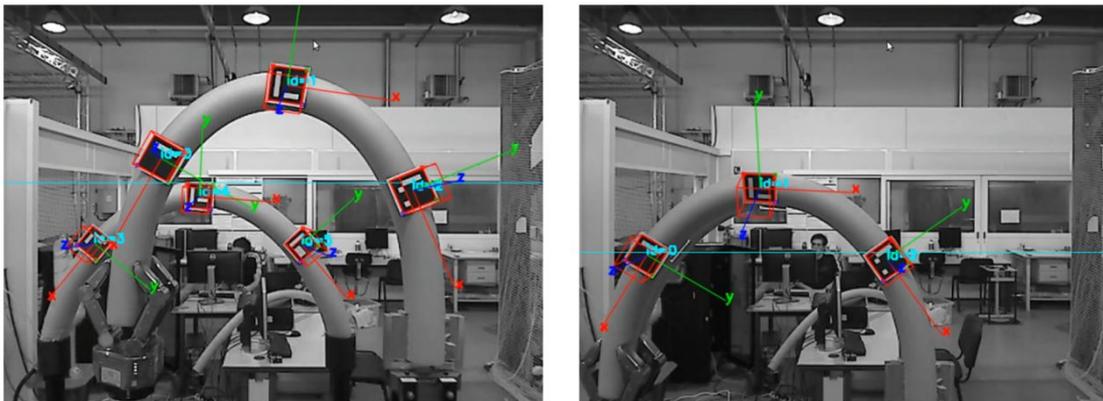


FIGURE 2.18: An experiment performed with the method proposed by [CCR21]. The image on the left shows the initial configuration of the object (back object) and the target (front object). The image on the right shows the final configuration of the object with respect to its target.

Recently, [CCR21] proposed an algorithm to manipulate a flexible beam using a dual-arm robot as shown in Fig. 2.18. In this technique, the object's deformation model is not required because it is experimentally constructed. Several markers were used to detect some points of the large beam and move them toward their targets. This method is used for small deformations since they used the local method to develop the proposed controller. [ZNF⁺18] proposed an algorithm to control the 2-D shape of an LDO (a wire) using two robotic manipulators. In this paper, the LDO's shape is estimated by a Fourier series. A least-squares minimization scheme is adopted to calculate the Jacobian matrix, which relates the motion of the end-effectors to the deformation of the LDO. Its hardware setup for LDO manipulation is illustrated in Fig. 2.19.

The main limitations of these methods are that they often can be used for low-speed movements and fixed targets. Furthermore, they cannot be used for large deformations, and since they are attached to the feedback of the used sensors, they are not very accurate.

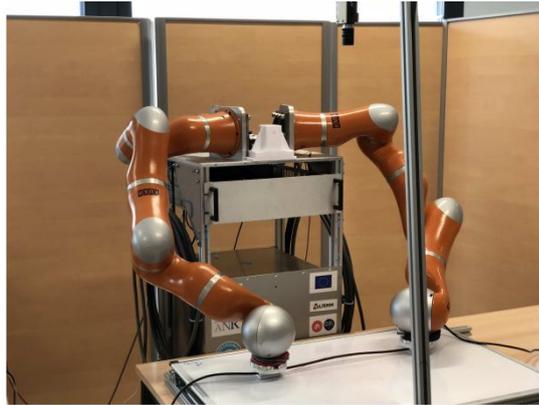


FIGURE 2.19: The experimental setup for LDO manipulation used in [ZNF⁺18]. In this work, a Fourier series is used to parameterize the shape. The shape parameters are used to estimate a local deformation model of the cable online. A velocity control rule is applied to the robot using the deformation model to deform the cable into the target.

2.4.1 Conclusion

In this chapter, we reviewed the literature on DO manipulation, and we investigated the existing works in this domain. As discussed, prior studies have not taken into account the deformation of the objects in the agriculture sector; and with their limitations, it is not easy to apply their strategies to agricultural activities. Some of the presented studies required much information regarding the objects. Some of them are not capable of handling large deformations. Another issue is that existing techniques cannot provide objects' dynamic properties online. However, when it comes to agricultural elements, the biggest problem is that there are many distinct items with varying physical attributes. Furthermore, the system should be able to handle the soft and delicate branches during the manipulating process with respect to their various shapes and sizes with large deformations. Thus, proposing novel methods to cover some of the existing challenges seems to be essential.

2.5 Positioning of this thesis with regards to the literature



FIGURE 2.20: A schematic of an LDO and the used robot (Campero) in the current research.

The main objective of this dissertation is to propose new control methods to allow the automatic robotic manipulation of LDOs. Our goal is to develop manipulation algorithms that can be used in agriculture in the future, as shown in Fig. 2.20. We use a mobile manipulator robot named Campero (Robotnik) with a Ur10 arm (Universal Robots) with six DOF. Robotic manipulators with six DOF allow operating the robot in any arbitrary position and orientation in 3-D space, making them appropriate for harvesting and pruning the trees [DBM⁺20].

To start designing the controllers, we adopt visual information as the feedback for our methods in view of the fact that visual sensors are sufficient to obtain the required information. They are cheap and non-invasive, and commonly used in industry and agriculture. Regarding control approaches, it is evident that a feedback signal is necessary in order to control the shape of an LDO precisely. In this thesis, we choose to present deformation techniques that are independent of any particular model since they can be used to manipulate more types of objects. Moreover, offering these kinds of techniques makes it easy to apply the same method to different objects. As mentioned, the existing methods require some initial steps such as obtaining objects' physical properties, which is not a necessary step in our methods. Unlike data-driven methods, we do not need to collect a lot of data beforehand. In addition, we intend to propose two new algorithms (in chapters 3 and 5) based on novel methods of computing the Jacobian matrix for linear objects that are easily computable. We also offer an adaptive controller that can be used with the least possible information about the object (in chapter 4, for controlling an arbitrary point and in chapter 6, for controlling the entire shape).

Chapter 3

An offline geometric method for LDOs deformation control without sensing the full shape

As a starting point of our research, we focus on [linear deformable objects \(LDO\)](#) manipulation, aiming to regulate the position of an arbitrary point (or a few points) along their length. We seek to find a solution to this problem without perceiving their entire shape during the deformation procedure. This can be helpful in applications where one needs to control only a part of the object, such as cutting wires and pruning the branches. Since servoing the entire shape is challenging and not trivial in some cases, it could be simpler only to track the controlled part in the mentioned tasks. Tracking a feature on the object is much more feasible since the detection algorithm can only focus on perceiving a specific feature, part, or mark on the object that we aim to control, and the entire possible workspace of the object's motion is not needed to be covered by the camera workspace. As a result, we seek a solution to this problem that does not involve tracking or online modeling of the complete object. To address the aforementioned issue, we attempt to propose a new approach to deforming linear objects with robots that takes these challenges into account. Specifically, we consider a fixed-length elastic linear object lying on a [2-D](#) workspace. Our main idea is to encode the object's deformation behavior in an offline constant Jacobian matrix. To derive this Jacobian, we use an offline geometric deformation modeling and based on this Jacobian, we then propose a robotic control law that is capable of driving a set of shape features on the object toward prescribed targets in [section 3.4](#). We illustrate the proposed approach in simulation and in experiments with real [LDOs](#) in [section 3.6](#).

3.1 Introduction

In the state of the art, successful methods have been proposed to manipulate [LDOs](#) based on a pre-existing deformation model [[BM14](#),[KFB⁺21](#),[SMBB20](#)] or on a model-free sensor-based adaptive scheme [[ZNF⁺18](#),[LKM20](#),[QMZ⁺21](#)]. An alternative approach based on a diminishing rigidity assumption has also been proposed [[Ber13](#)]. At run-time, the cited methods that perform closed-loop control require sensing the full shape of the object (in order to simulate the deformation model), or running adaptive schemes based on a Jacobian matrix to estimate the deformation behavior. Another solution is to address [LDO](#) manipulation problems via the use of machine learning that has gained popularity recently [[YZJB20](#),[HHS⁺19](#),[YZL22](#),[LK21](#)]. The main challenges of these methods are that they are usually complex and require a lot of effort in training.

We discussed the literature on LDO manipulation in chapter 2. The fundamental objective of the current chapter is to propose an approach that does not have the drawbacks of the previous works [LKM20, QMZ⁺21, KFB⁺21, SMBB20, Ber13, ARM⁺20] that are required sensing the full shape of the object and simulating the deformation model at run-time. We would like to present an approach which works by computing offline a model (a deformation Jacobian) based on geometric modeling and assuming an elastic behavior (we assume the types of objects that we study in this dissertation do not have any plastic deformation). As mentioned in the previous chapter, the Jacobian matrix is a matrix that links the motion of the robot to the changes in the deformation of the object. At run-time, we desire to present a method that needs to sense the values of the servoed features only, and not the full shape. To develop our model, we build on recent techniques for DO manipulation [KFB⁺21] and multirobot systems control [ALNM21]. It is interesting to use this approach in a very different application (i.e., manipulation of DOs) since by using this method, we are able to propose a controller without calculating a deformation model or tracking the complete shape of the object during the deformation process. Our model is geometric and thus we also follow along the lines of recent work [ARM⁺20, SBAMO22] which proposed the robotic control of the shape of DOs using geometric deformation modeling techniques [SA07, MHHR07b]. A significant advantage of geometric models is that they are not attached to the dynamic properties of a specific object. Therefore, using a geometric model to propose a controller helps us to easily adopt the same method to manipulate a broader range of objects.

3.2 Our contributions

The contributions of this chapter are:

- A novel method for 2-D control of the shape of elastic linear objects based on an offline geometric model which, at run-time, does not require computing a deformation model or sensing the full shape of the object.
- A novel link between prior work in two different domains: DO modeling and multi-robot systems control. This link is a promising starting point for potential future developments in the same direction.
- A practical and straightforward method to control LDOs based on an offline model that does not need to know any physical or material parameter or characterization of the object's deformation. We only need the knowledge of all the nodes that construct the shape of the object at rest shape so as to compute the Jacobian matrix.
- Successful validation of the applicability of the proposed method, in simulation and via robotic experiments with diverse objects.

The envisioned applications of the proposed approach are diverse, including for instance plant pruning tasks in agriculture, or assembly operations in industrial contexts.

3.3 Problem formulation

The problem we address is the shape control of a linear elastic object by means of robotic manipulation. Our goal is to control the position of one or multiple features on the object. We divide the shape of the LDO into a set of nodes. p_1 and p_2 indicate the position of the center of each node in the camera frame, and p_3 indicates its angle, as shown in Fig. 3.1. To clarify, p_3 is the tangent angle to the object (normal to the object) at the center of each node, and we use the edge position (the top and bottom) of each node to compute it. The problem we are dealing with has been given the name of shape servoing

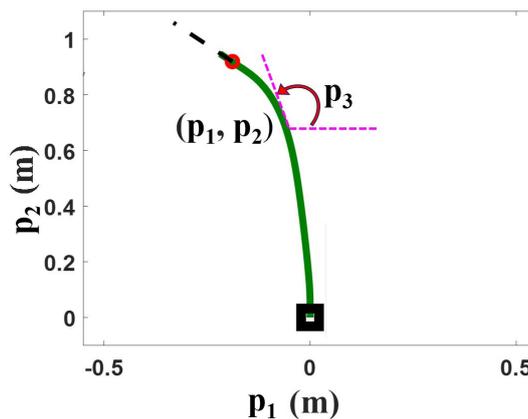


FIGURE 3.1: The studied object in this dissertation. p_1 and p_2 represented the position of the LDO and p_3 shows its angle.

in the literature [NAL17, QMZ⁺21, SBAMO22]. Specifically, we consider the following conditions and assumptions:

- The object is an LDO and lies in a 2-D workspace. It has fixed length and it deforms elastically (not plastically). The shape of the object at rest (i.e., the object's shape without applying any manipulation or deformation) is known. The object's shape always remains stable (i.e., in quasi-static equilibrium) during its manipulation.
- One or multiple robots (i.e., grippers) grasp the object. The velocity of each gripper can be set by a robotic controller. Each gripper constrains a specific gripped region on the object. There may also be anchored regions on the object, i.e., whose position is constant due to external constraints (e.g., a grounded region). The gripped and anchored regions of the object are known before starting the task, and they remain fixed throughout the execution. The grasping configuration used is suitable for completing the task described next.
- The addressed task consists in controlling a set of features that encode information about the object's shape (e.g., the 2-D positions of certain parts of the object). Each feature has a desired value, which is fixed and prescribed before starting the task. The current values of the features can be continually sensed at run-time (e.g., with vision). Our objective is to find a closed-loop control law for the velocity of every gripper so that every feature is driven to its desired value.

Fig. 3.2 shows the block diagram of the method we propose to carry out this task. We describe the method next.

3.4 Proposed LDO modeling and shape control

We represent the LDO discretely with a set of n nodes, indexed $1, \dots, n$. The nodes are sampled uniformly and consecutively along the object's length. The object's rest shape, \mathbf{c} , is the stacking of the positions at rest of all nodes: $\mathbf{c} = [\mathbf{c}_1^\top, \mathbf{c}_2^\top, \dots, \mathbf{c}_n^\top]^\top \in \mathbb{R}^{2n}$. We group the nodes in sets of three. We call each of these sets a triad, as shown in 3.3. The full linear object has w chained triads, $\mathcal{T}_k = \{k, k+1, k+2\}$ (where the indexes are taken *mod* n) for $k = 1, \dots, w$. We consider that the object's rest shape is open; e.g., a rod with two ends. Therefore, $w = n - 2$. We denote the current shape of the object (i.e., the current mesh node positions) as $\mathbf{q} = [\mathbf{q}_1^\top, \mathbf{q}_2^\top, \dots, \mathbf{q}_n^\top]^\top \in \mathbb{R}^{2n}$.

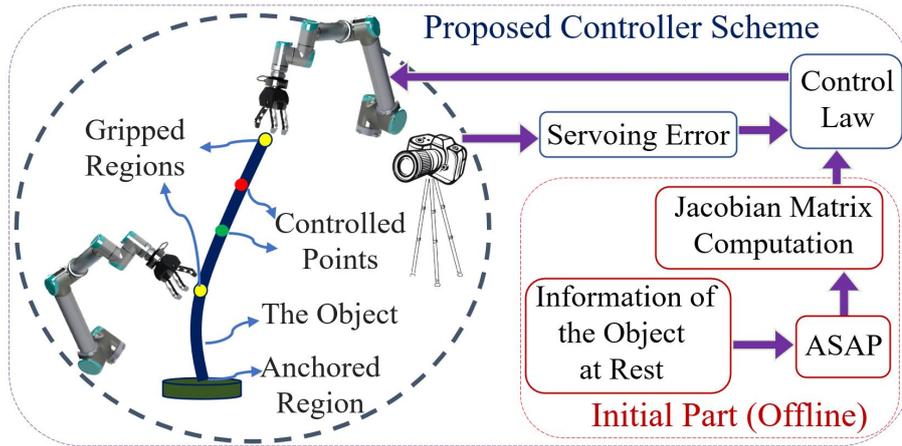
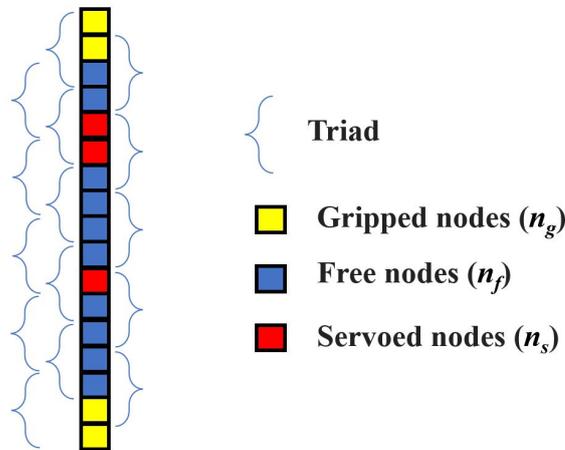


FIGURE 3.2: Schematic diagram of the proposed method.

FIGURE 3.3: The LDO is discretized into a set of nodes (n_g gripped nodes, n_s servoed nodes, and n_f free nodes). The nodes are grouped in sets of three, which we call each of these sets a triad.

Our approach is based on an **As-Similar-As-Possible (ASAP)** modeling. **ASAP** is a way to model deformations that has been used in the computer graphics domain for visualization applications [IMH05, LZ⁺08, CG16]. It is closely related to, and sometimes used in conjunction with, the popular model **ARAP (As-Rigid-As-Possible)** [IMH05, SA07, LG14, SBAMO22], which has been shown to be effective in modeling the deformation of deformable objects. **ASAP** is based on the intuitive idea that every local region of the modeled object has a tendency to preserve its original shape up to translation, rotation and scaling: i.e., a similarity transformation of the original shape, hence the name of the method. In concrete terms, **ASAP** is formulated as a deformation energy, and the quasi-static configurations of the object's shape are local minima of that energy.

There are different formulations of **ASAP/ARAP**; all are based on the same underlying concepts, but they use different definitions of the local regions. In particular, an **ASAP** energy in 2-D was proposed in [ALNM21], in a context not related with **DOs**: specifically, the proposed energy was used as the cost function in a distributed controller of a multi-robot formation. In [ALNM21], the elements forming the triads were robots, whereas they are the points on the object's body in our case. As we show next, the formulation of [ALNM21] is compact and adapts perfectly to our needs, which motivates our use of

it. The energy proposed in [ALNM21] is a sum of energies over every triad, as follows:

$$E_a = \sum_{k=1}^w E_k, \quad E_k = \frac{1}{2} \sum_{i \in \mathcal{T}_k} \|(\mathbf{q}_i - \mathbf{q}_{0_k}) - \mathbf{H}_k(\mathbf{c}_i - \mathbf{c}_{0_k})\|^2 \quad (3.1)$$

\mathbf{q}_{0_k} and \mathbf{c}_{0_k} are the centroids of the current and rest positions of the three nodes in \mathcal{T}_k . $\mathbf{H}_k \in \mathbb{R}^{2 \times 2}$ is the least-squares similarity (rotation and uniform scaling) transformation between these two sets of positions. E_a can be expressed compactly:

$$E_a = -\frac{1}{2} \mathbf{q}^\top \mathbf{A} \mathbf{q} \quad (3.2)$$

In this expression, \mathbf{A} is a constant matrix that encapsulates the deformation (in the ASAP sense) of the object. \mathbf{A} is a sparse symmetric matrix based on the triad structure of the mesh and on its shape at rest, \mathbf{c} . Note that this compact 2-D formulation cannot be directly extended to 3-D, where non-linearities appear (see, e.g., [SCOL+04]). An expression of matrix \mathbf{A} was given for general triad structures and general shapes at rest in [ALNM21]. Next, starting from that expression, we give the particular form of the matrix in the case of our LDO model.

Specifically, we define $\mathbf{S} = [(0,1)^\top, (-1,0)^\top]$, i.e., a counterclockwise rotation of $\pi/2$ rad, and $\mathbf{T} = \mathbf{I}_n \otimes \mathbf{S}$, where \otimes denotes the Kronecker product. Then, $\mathbf{A}, \mathbf{A}_k, \mathbf{L}_k \in \mathbb{R}^{2n \times 2n}$, $\mathbf{L}_{gk} \in \mathbb{R}^{n \times n}$ are as follows:

$$\mathbf{A} = \sum_{k=1}^w \mathbf{A}_k, \quad \mathbf{A}_k = \frac{\mathbf{L}_k(\mathbf{c}\mathbf{c}^\top + \mathbf{T}\mathbf{c}\mathbf{c}^\top\mathbf{T}^\top)\mathbf{L}_k}{\mathbf{c}^\top\mathbf{L}_k\mathbf{c}} - \mathbf{L}_k \quad (3.3)$$

$$\mathbf{L}_k = \mathbf{L}_{gk} \otimes \mathbf{I}_2, \quad \mathbf{L}_{gk}[i,j] = \begin{cases} 2/3, & \text{if } i = j \text{ \& } i \in \mathcal{T}_k \\ -1/3, & \text{if } i \neq j \text{ \& } i, j \in \mathcal{T}_k \\ 0 & \text{otherwise} \end{cases}$$

ASAP (and ARAP) are models that represent physical behavior at the geometric level, rather than at the mechanical one. Still, these models assume the object behavior is quasi-static, and they are based on formulating a deformation energy. In these aspects they are analogous to elastic FEM models [DBPC18, KFB+21]. Therefore, it is possible to compute equivalent mechanical magnitudes (e.g., forces) in ASAP or ARAP. We will therefore use an analysis similar to [DBPC18, KFB+21] to derive a control law from ASAP.

The first step is to know the forces at the nodes due to the ASAP energy. By definition these forces are equal to the negated gradient of the energy. Therefore, noting that \mathbf{A} is symmetric, the nodal forces $\mathbf{f}_a \in \mathbb{R}^{2n}$ are:

$$\mathbf{f}_a = -\frac{\partial E_a}{\partial \mathbf{q}} = \mathbf{A} \mathbf{q} \quad (3.4)$$

It is interesting to note that this is a linear expression in \mathbf{q} . We compute its time derivative:

$$\mathbf{A} \dot{\mathbf{q}} = \dot{\mathbf{f}}_a \quad (3.5)$$

From (3.5), we will next derive a control law following a similar strategy to the one used in [KFB+21] with a FEM model. We first define a partition of the nodes. Concretely, we divide the set of nodes into n_g gripped nodes, n_s servoed nodes, and n_f free nodes, such that $n_g + n_s + n_f = n$, as shown in Fig. 3.3. The servoed nodes are the ones we track and control, whereas free nodes are those we do not regulate. Note that the set of gripped nodes includes all regions of the object whose position is constrained externally: i.e., the grasped regions, and also the anchored regions. The positions of the nodes are denoted,

respectively, by $\mathbf{q}_g \in \mathbb{R}^{2n_g}$, $\mathbf{q}_s \in \mathbb{R}^{2n_s}$, $\mathbf{q}_f \in \mathbb{R}^{2n_f}$. Matrix \mathbf{A} and the vector of forces \mathbf{f}_a are also partitioned accordingly, and hence (3.5) takes the form:

$$\begin{pmatrix} \mathbf{A}_{gg} & \mathbf{A}_{gs} & \mathbf{A}_{gf} \\ \mathbf{A}_{sg} & \mathbf{A}_{ss} & \mathbf{A}_{sf} \\ \mathbf{A}_{fg} & \mathbf{A}_{fs} & \mathbf{A}_{ff} \end{pmatrix} \begin{pmatrix} \dot{\mathbf{q}}_g \\ \dot{\mathbf{q}}_s \\ \dot{\mathbf{q}}_f \end{pmatrix} = \begin{pmatrix} \dot{\mathbf{f}}_{ag} \\ \dot{\mathbf{f}}_{as} \\ \dot{\mathbf{f}}_{af} \end{pmatrix} \quad (3.6)$$

As the object is always in quasi-static equilibrium, all resultant forces (i.e., sum of ASAP force and external force) on all nodes are always zero. Hence, their time-derivative is zero too. Therefore, using subscript *ex* to denote external forces, we have:

$$\dot{\mathbf{f}}_{ag} + \dot{\mathbf{f}}_{exg} = 0, \quad \dot{\mathbf{f}}_{as} + \dot{\mathbf{f}}_{exs} = 0, \quad \dot{\mathbf{f}}_{af} + \dot{\mathbf{f}}_{exf} = 0 \quad (3.7)$$

The gripped nodes are subjected to the external forces that cause the object to deform, which are in general time-varying forces. Therefore, $\dot{\mathbf{f}}_{exg}$ are not zero, and consequently $\dot{\mathbf{f}}_{ag}$ is not zero. On the other hand, the nodes that are not being gripped (i.e., the servoed nodes and free nodes) are subjected to constant external forces (since we assumed that the object's shape always remains in quasi-static equilibrium). Typically these constant external forces are zero, or equal to the gravity force. This means $\dot{\mathbf{f}}_{exs} = 0$ and $\dot{\mathbf{f}}_{exf} = 0$. Hence, from (3.7), $\dot{\mathbf{f}}_{as} = 0$ and $\dot{\mathbf{f}}_{af} = 0$. Using these latter two conditions in (3.6), we get:

$$\begin{pmatrix} \mathbf{A}_{sg} & \mathbf{A}_{ss} & \mathbf{A}_{sf} \\ \mathbf{A}_{fg} & \mathbf{A}_{fs} & \mathbf{A}_{ff} \end{pmatrix} \begin{pmatrix} \dot{\mathbf{q}}_g \\ \dot{\mathbf{q}}_s \\ \dot{\mathbf{q}}_f \end{pmatrix} = 0 \quad (3.8)$$

From (3.8), we can obtain the key expression in our development: the relation between gripped and servoed node motions. This expression is as follows:

$$\dot{\mathbf{q}}_s = \mathbf{J}_{sg} \dot{\mathbf{q}}_g \quad (3.9)$$

with $\mathbf{J}_{sg} = -(\mathbf{A}_{ss} - \mathbf{A}_{sf} \mathbf{A}_{ff}^{-1} \mathbf{A}_{fs})^{-1} (\mathbf{A}_{sg} - \mathbf{A}_{sf} \mathbf{A}_{ff}^{-1} \mathbf{A}_{fg})$. Similarly to [IMH05, KFB⁺21, YZL22], we assume the matrices that have to be inverted are full-rank. This comes from the fact that the shape is completely constrained by the gripped nodes.

3.4.1 Control law

Assuming the desired positions for the servoed nodes are $\mathbf{q}_{sd} \in \mathbb{R}^{2n_s}$, we define the servoing error as:

$$\mathbf{e}_s = \mathbf{q}_s - \mathbf{q}_{sd} \quad (3.10)$$

Now, from (3.9) and (3.10), we can propose the following proportional control law:

$$\mathbf{u}_r = -k_r \mathbf{J}_{sg}^\dagger \mathbf{e}_s \quad (3.11)$$

where k_r is a positive gain, \dagger denotes the pseudoinverse, and \mathbf{u}_r is the velocity to be applied to the gripped nodes: $\mathbf{u}_r = \dot{\mathbf{q}}_g$.

Frequently, the gripper used in real applications can constrain not only the position of the gripped point of the linear object, but also the orientation (i.e., the line tangent to the object's curve) at that point. Interestingly, this can be introduced in a straightforward way in our model, as follows. For simplicity, assume a gripper h grips two (adjacent) nodes. Calling the positions of these nodes $\mathbf{g}_1 = [g_{1p1}, g_{1p2}]^\top$ and $\mathbf{g}_2 = [g_{2p1}, g_{2p2}]^\top$, the three DOF configuration of the gripper in the global frame is $[p_{1g}, p_{2g}, p_{3g}]^\top = [(g_{1p1} + g_{2p1})/2, (g_{1p2} + g_{2p2})/2, \text{atan2}(g_{2p2} - g_{1p2}, g_{2p1} - g_{1p1})]^\top$ (p_1 , p_2 , and p_3 show the state

of any point on the object, as shown in Fig. 3.1). Denoting the gripped length (i.e., length of the line segment between \mathbf{g}_1 and \mathbf{g}_2) by l_g , a Jacobian matrix block that maps gripper h 's velocities to the velocities of its gripped nodes is as follows:

$$\mathbf{J}_{gph} = \begin{pmatrix} 1 & 0 & (l_g/2) \sin p_{3g} \\ 0 & 1 & -(l_g/2) \cos p_{3g} \\ 1 & 0 & -(l_g/2) \sin p_{3g} \\ 0 & 1 & (l_g/2) \cos p_{3g} \end{pmatrix} \quad (3.12)$$

Assuming there are n_e such grippers, we assemble the Jacobian blocks for all of them in a full Jacobian matrix \mathbf{J}_{gp} of size $4n_e \times 3n_e$. We also stack the gripper velocities (two linear and an angular velocities) in a vector \mathbf{u}_g of length $3n_e$. Then we define the control law:

$$\mathbf{u}_g = -k_g \mathbf{J}_{sp}^+ \mathbf{e}_s \quad (3.13)$$

where $\mathbf{J}_{sp} = \mathbf{J}_{sg} \mathbf{J}_{gp}$ is the Jacobian relating servoed nodes to grippers. A gain matrix can be used in place of k_g , to weight differently translation and rotation velocities.

3.4.2 Discussion

We make several remarks about the presented idea.

- For elastic linear objects, there are well-known instabilities at certain configurations where the shape changes non-smoothly [BM14]. Our formulation is only valid in the regions of the object's shape space where the changes are stable and smooth, i.e., where a relation such as (3.9) can be defined.
- In [KFB⁺21], a matrix \mathbf{K} that plays an analogous role to our \mathbf{A} is used. That \mathbf{K} is the tangent stiffness matrix of the FEM structure that represents the object, and it is a function of \mathbf{q} . This means one needs to estimate \mathbf{q} (i.e., the full shape of the object) at run-time to compute the control law, even if only a few of the nodes are servoed and most are not. We avoid this requirement since our matrix \mathbf{A} is constant and can be computed offline. This is important because measuring the shape of DOs robustly is challenging.
- \mathbf{A} and then \mathbf{J}_{sg} can be computed from the rest shape of the object (which is direct to know for a straight elastic rod) and the knowledge of what nodes are gripped and what nodes are to be servoed. At run-time, our control law (3.11) only needs to compute \mathbf{e}_s , which requires measuring the positions of the servoed nodes only (not the shape of the full object).
- Even if our Jacobian is computed using the rest shape, this does not mean that it is only valid at the rest shape. On the contrary, it is the valid and exact Jacobian of the object, under the ASAP modeling, at every shape that satisfies our prior assumptions (stable deformation).
- Recent works performed successful shape control with approximated deformation models (FEM with imprecise parameters [KFB⁺21], ARAP [SBAMO22]). By using the measurement of the object's shape in a feedback loop, the control laws in these works are convergent even if the deformation model is only coarsely accurate. This justifies our use of ASAP.
- Our controller idea directly accommodates multiple robots (grippers) without any change in the methodology. Just by knowing which nodes are gripped by each gripper, one can derive directly the control laws for all grippers.
- However, in order to use the described idea to control the shape of real-world objects, there remains a major issue: in general, ASAP is not a good model of the deformation of real-world elastic objects. We solve this issue by identifying specific conditions under which it becomes a good model, as explained in the next section.

3.5 ASAP as an ARAP approximation

Although **ASAP** is a proper method to calculate the deformation Jacobian of elastic objects, it has some limitations. For instance, it is not a good representation of the behavior of elastic objects because it does not preserve their physical size. In general, an object simulated using **ASAP** can easily shrink or grow in size unrealistically [IMH05, LZX⁺08]. On the other hand, **ARAP** is a good model of deformation behaviors of elastic objects [SA07, LG14]. **ARAP** tends to preserve rigidity locally and by doing so, it keeps the size of the object under control, unlike **ASAP** does. However, **ARAP** is a nonlinear model, and its Jacobian depends on the current shape \mathbf{q} (it should be mentioned that **ARAP** is more general than **ASAP**, and we will use it (**ARAP**) in chapter 5 to propose an optimal controller). Avoiding this dependency is precisely why we want to use **ASAP**, which is simpler and faster than **ARAP**.

Our triad-based formulation can be used both for **ASAP** and **ARAP**. For both models, an essential element is the computation of an optimal transformation for each triad. In **ASAP**, this transformation is a similarity (rotation and scaling), as described above: \mathbf{H}_k , for triad k . In **ARAP**, the transformation is a pure rotation: \mathbf{R}_k , for triad k . These transformations satisfy:

$$\mathbf{H}_k = s_k \mathbf{R}_k \quad \forall k \quad (3.14)$$

Therefore, we can make the observation that if the condition $s_k = 1 \quad \forall k$ holds, **ASAP** and **ARAP** are equivalent. Motivated by this observation, we next show that this condition approximately holds in the scenario we consider.

3.5.1 Geometric analysis for a triad

Our geometric modeling of a generic triad, shown in Fig. 3.4, is motivated next. This modeling assumes the rest shape of every triad is a straight line. Note that our modeling of the object is discrete. Hence, the shape of the object between two adjacent nodes is represented as a straight line segment. Recall that the object's length (equal to the sum of the segments' lengths in our modeling) is fixed and we have divided it into meshes of the same shape and size. Therefore all segments always have equal and fixed lengths (l). The change of curvature along the object is represented via the angle θ relative to r , which is the line tangent to the object's curve according to our discrete modeling. Due to the symmetry of the layout, the variables of the **ASAP/ARAP** models can be obtained via direct trigonometric relations. In particular, we can obtain:

$$s = \cos(\theta) \quad (3.15)$$

Our goal was to have values of s close to 1 (3.14): it is clear from (3.15) that this will happen when θ is small. Therefore, we want θ to be always small for all triads which implies a smooth (not sudden) deformation along the object. Note that θ is always between $-\pi/2$ and $\pi/2$, which means that the $\cos(\theta)$ is always positive.

Notice that θ is small if the local curvature at every point is small: this will typically be true if n is high enough to sample the object densely, and there are no extreme local deformations (i.e., very high bending at some point along the rod). Observe that the global deformation of the object can still be large even if the local curvature is small at every one of its points, as shown in our experiments (section 3.6).

We used the gradients (which express the forces) of the **ASAP** energy to derive our controller. To support the use of this controller, next we study how well the **ASAP** gradients can approximate the **ARAP** ones. In Fig. 3.4, the **ASAP** gradient for the shown triad

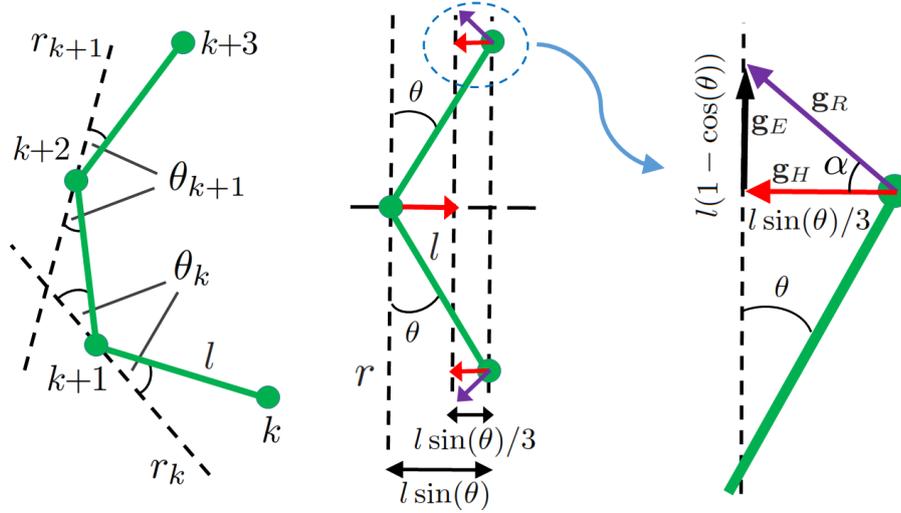


FIGURE 3.4: Geometric modeling. Left: four nodes in two triads $\mathcal{T}_k, \mathcal{T}_{k+1}$, with their angles θ_k, θ_{k+1} . Center: a generic triad. Right: close-up of a region of the center plot. Nodes are shown as circles, gradient vectors as arrows.

is denoted by \mathbf{g}_H , the **ARAP** gradient as \mathbf{g}_R , and the approximation error as $\mathbf{g}_E = \mathbf{g}_R - \mathbf{g}_H$. Due to the properties of least-squares alignment, the three endpoints of the gradient vectors form the rest (straight) shape and their centroid is the same as the nodes' centroid. Moreover, the vectors have certain symmetries due to the symmetry of the layout (the nodes form an isosceles triangle). We can obtain the following expression:

$$\alpha = \arctan \frac{\|\mathbf{g}_E\|}{\|\mathbf{g}_H\|} = \arctan \frac{3(1 - \cos(\theta))}{\sin(\theta)} \quad (3.16)$$

As $|\theta| \leq \pi/2$ rad by definition, (3.16) implies that $|\alpha| < \pi/2$ rad. This means that the inner product of the true gradient vector (\mathbf{g}_R) and the one we use as its approximation (\mathbf{g}_H) is always non-negative. In gradient-based algorithms, this is usually a sufficient condition for obtaining the desired performance (e.g., in gradient descent). Note that we do not need to study the center node of the triad, because for that node $\mathbf{g}_H = \mathbf{g}_R$, i.e., the approximation is perfect.

Overall, this geometric analysis supports the appropriateness of approximating elastic behavior with an **ASAP** model in our scenario. Note that this is only a partial analysis restricted to one triad; for each point, its total gradient is the sum of gradients over all the triads the point belongs to.

3.6 Experimental validation

We test the approach in simulation using **KER** and **ARAP** model and in hardware experiments using various real objects having diverse characteristics, in different scenarios¹.

3.6.1 Simulation

We use Matlab and the models we explained in chapter 2 (**ARAP** and **KER**) to test the presented scheme in simulations. As mentioned in the previous chapter, we only use these models to model the object in simulation in order to verify the methods we present

¹A video of some of our simulations and experimental results can be found at: <https://www.youtube.com/watch?v=LJ8jDTFPIR8>

in this dissertation (as shown in Fig. 2.8). We assume $n = 50$, and we control different points along the length of the various simulated objects. It should be noted that for other similar values of n , similar results are obtained. To manipulate the object, (3.13) is implemented to the used grippers. The value of s for all triads always stayed between 0.995 and 1, which is consistent with our analysis in section 3.5.1.

In the plots of this section, the error related to the position of each controlled point is defined from (3.10) as follows (l indicates the controlled point number):

$$error_l = \sqrt{(p_{1s-l} - p_{1sd-l})^2 + (p_{2s-l} - p_{2sd-l})^2} \quad (3.17)$$

where $[p_{1s-l}, p_{2s-l}]^T$ is the current position of the controlled point and $[p_{1sd-l}, p_{2sd-l}]^T$ is its desired position.

3.6.2 Robotic experiments with real objects

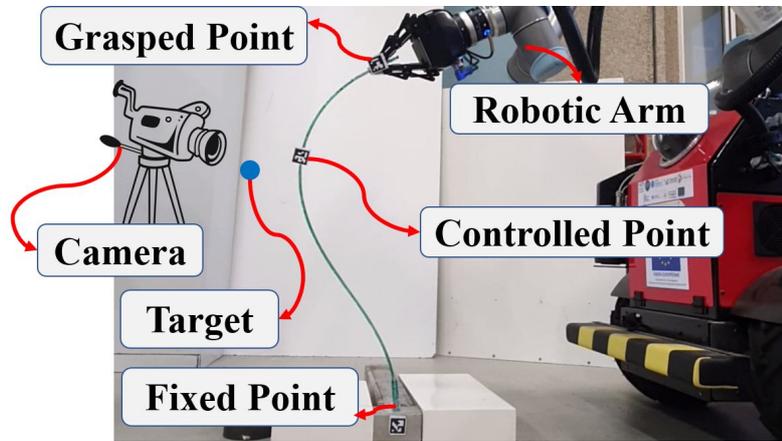


FIGURE 3.5: The experimental setup used in this chapter. Control point(s) and the grasped point are tracked with a camera.

The proposed method with the offered controller (3.13) is implemented in a robotic manipulation system to study its performance, as shown in Fig. 3.5. The experiments are performed with a UR10 arm. Fifty points are used to discretize the object (i.e., $n = 50$). Other values of n in a similar range (several tens) produce similar results. The controlled points (servoed nodes) and the gripped points are tracked with a fixed Logitech C270 camera, and ArUco markers have been used to obtain the pose of these points using the OpenCV library of C++. It's worth noting that utilizing markers to detect objects is a well-known and commonly utilized technique [CFL⁺20, ARM⁺20, GJMSMCMJ14]. The used marker should be detectable by cameras and provide an appropriate estimation of the position of the points.

To validate the performance of the proposed method, various experiments are performed to control one, two, and three different points along the length of the objects. We use three different LDOs with various characteristics to conduct the tests: a thin flexible rod made of plastic with the length of $0.87(m)$, a deformable foam rod with the length of $0.87(m)$, and a small deformable foam rod with the length of $0.26(m)$. Note that by selecting an appropriate n , the object's length does not affect the controller's performance. The Jacobian matrix is obtained using the proposed method offline in section 3.4, and then, the states of the gripped points are updated using control law (3.13) at each instant.

3.6.3 Results of controlling one point

To investigate the controller's performance, in the first step, we try to control pose (position, and angle of the line tangent to the object's curve) of a single point along the object's length. To do that, the pose of the point is obtained using the position of its two closest nodes that we use to obtain the control law (3.13) with the defined Jacobian in section 3.4. Several simulations and experiments have been performed to validate the accuracy of the proposed algorithm, and the results are shown in Fig. 3.6 to Fig. 3.11.

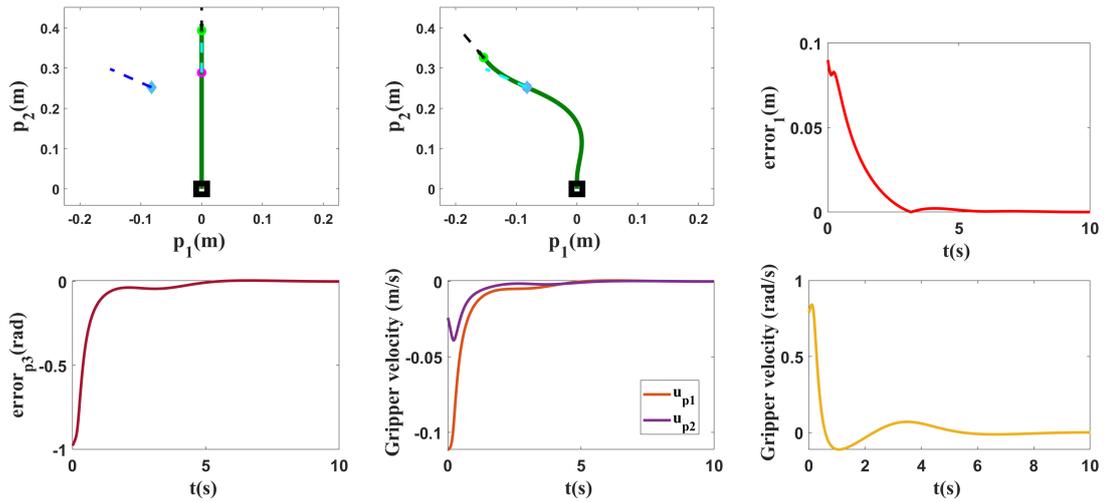


FIGURE 3.6: Simulation 3.1. Pose control of one point. The object is modeled using ARAP ($L = 0.41(m)$). From the top-left, the initial and final shapes, the errors, and the gripper velocities on different axes are shown. The controlled point can be found as a red circle with a dashed line segment used to show its angle. The target point can be found as a blue diamond with a dashed line segment used to show the target angle.

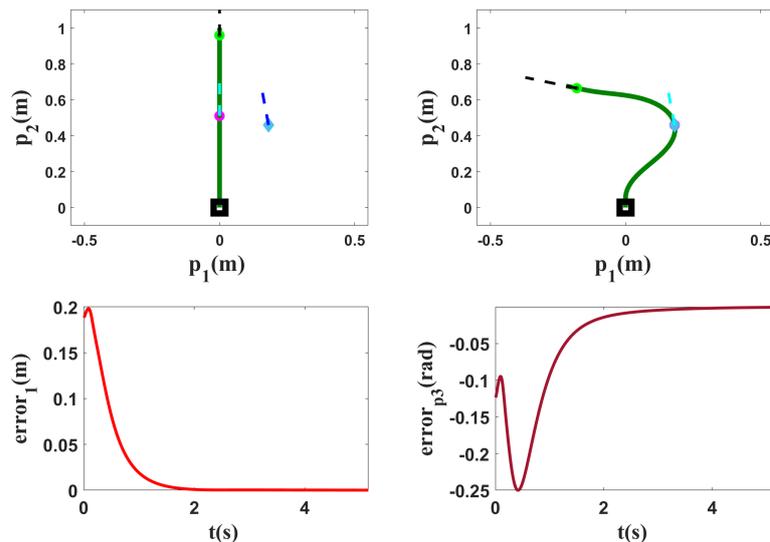


FIGURE 3.7: Simulation 3.2. Pose control of one point. The object is modeled using ARAP ($L = 1.00(m)$). The initial and final shapes can be seen in the first row. The controlled point can be found as a red circle with a dashed line segment used to show its angle. The target point can be found as a blue diamond with a dashed line segment used to show the target angle. The errors are plotted in the second row.

As one can see in Fig. 3.6 to Fig. 3.11., by applying the controller (3.13) during the deformation process, the errors of the controlled points converge to zero. In these tests, to show the performance of the proposed method, we define the error of the angle $error_{p_3} = \mathbf{p}_{3_{s-l}} - \mathbf{p}_{3_{sd-l}}$, where $\mathbf{p}_{3_{s-l}}$ and $\mathbf{p}_{3_{sd-l}}$ are the current and target value of the angle of the point, respectively. It is assumed that the angle of the object remains in the same range in each iteration. The initial, target, and final pose of the controlled points in all experiments are presented in Table 3.1. Moreover, the final errors of all tests (the absolute value of the difference between the final states and their targets) are also shown in that table.

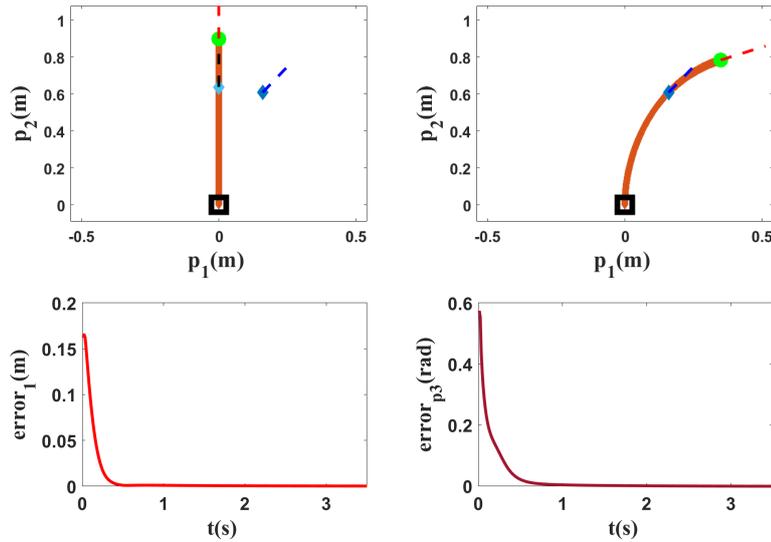


FIGURE 3.8: Simulation 3.3. Pose control of one point. The object is modeled using KER ($L = 0.90(m)$). The initial and final shapes can be seen in the first row. The target point can be found as a blue diamond with a dashed line segment used to show the target angle. The errors are plotted in the second row.

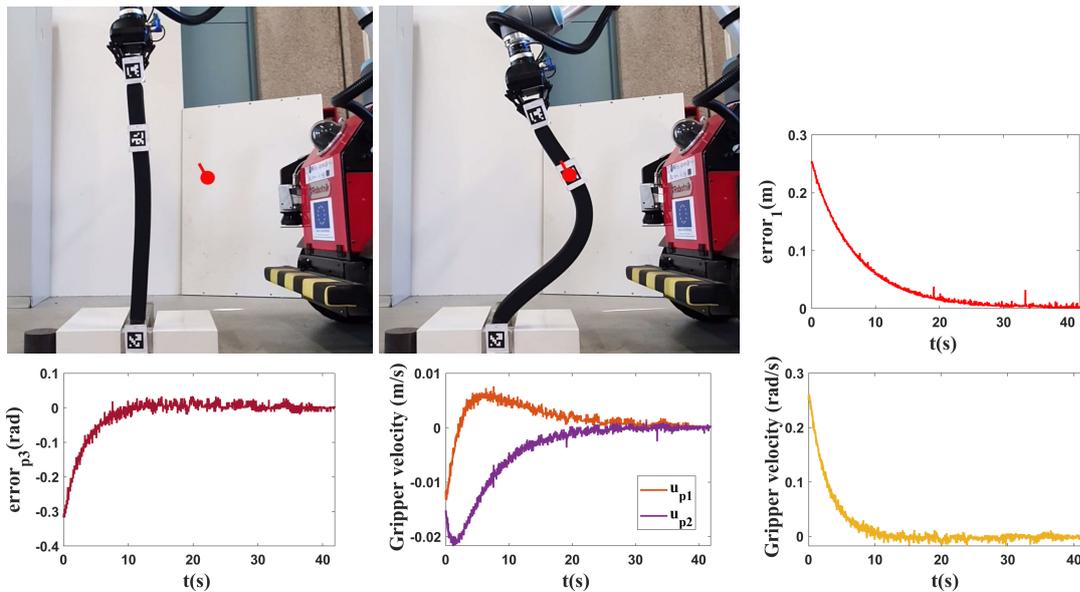


FIGURE 3.9: Experiment 3.1. Pose control of one point. From the top-left, the initial and final shapes, the errors, and the gripper velocities on different axes are shown. The target point can be found as a red circle with a dashed line segment used to show the target angle.

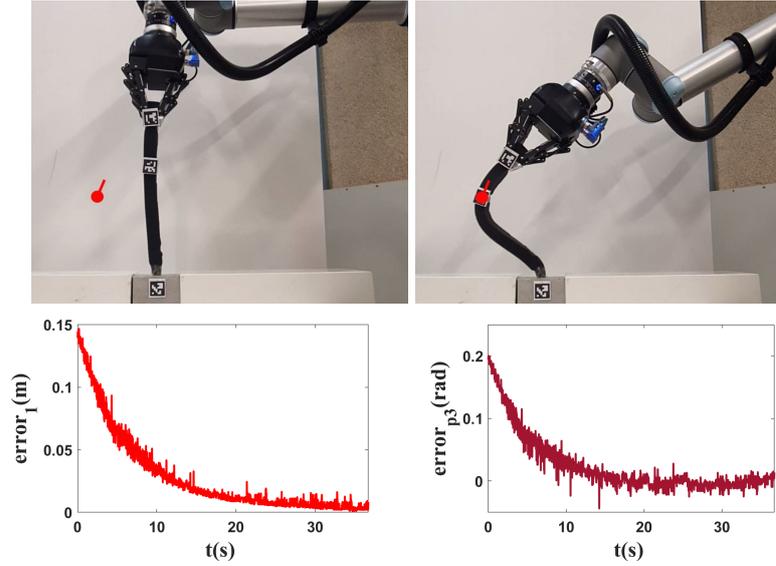


FIGURE 3.10: Experiment 3.2. Pose control of one point. The initial and final shapes can be found in the first row. The errors are plotted in the second row.

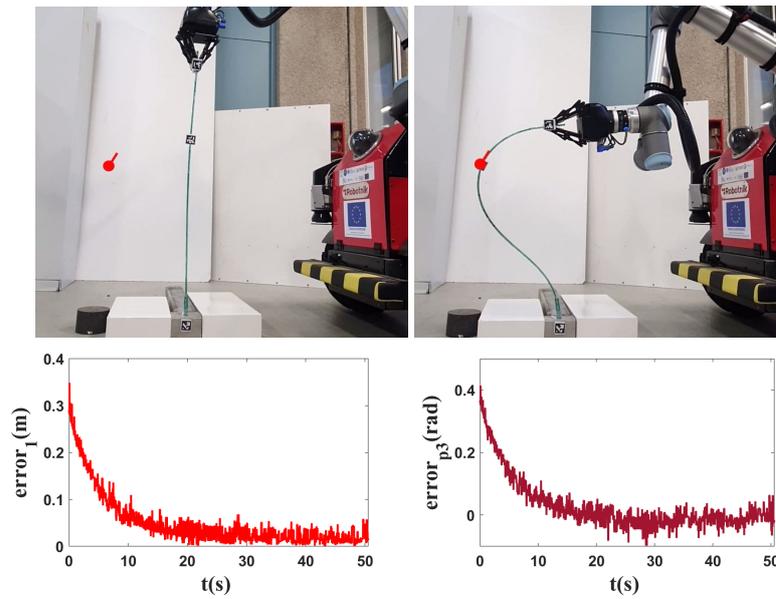


FIGURE 3.11: Experiment 3.3. Pose control of one point. The initial and final shapes can be found in the first row. The errors are plotted in the second row.

TABLE 3.1: State of the controlled points in the performed tests for controlling one point.

Test number	Initial pose (m, m, rad)	Target pose (m, m, rad)	Final error (m, m, rad)
Sim. 3.1	[0.000, 0.291, 1.571]	[-0.086, 0.254, 2.547]	[0.000, 0.00, 0.000]
Sim. 3.2	[0.000, 0.510, 1.571]	[0.181, 0.459, 1.695]	[0.000, 0.00, 0.001]
Sim. 3.3	[0.000, 0.639, 1.571]	[0.159, 0.609, 0.997]	[0.000, 0.00, 0.001]
Exp. 3.1	[0.086, 0.650, 1.609]	[0.296, 0.502, 1.915]	[0.002, 0.003, 0.005]
Exp. 3.2	[-0.027, 0.266, 1.467]	[-0.146, 0.198, 1.258]	[0.001, 0.001, 0.019]
Exp. 3.3	[0.019, 0.678, 1.571]	[-0.283, 0.551, 1.144]	[0.003, 0.005, 0.009]

3.6.4 Results of controlling multiple points

In the next step, we take another pace and try to control position of two and three non-adjacent points using the proposed method.

In the initial step, we attempt to control two points using two different grippers. The results can be seen in Fig. 3.12 and Fig. 3.13. As one can be seen, the method is able to perform the defined tasks accurately. Table 3.2 displays the initial and target states of the controlled points as well as the final errors of these tests.

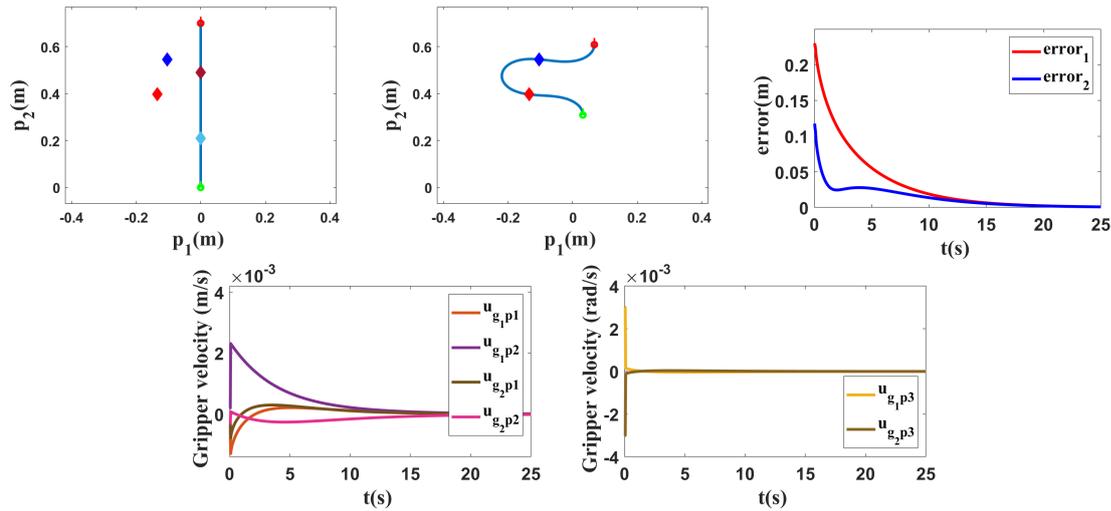


FIGURE 3.12: Simulation 3.4. Position control of two points using two grippers. The object is modeled using KER ($L = 0.70(m)$). The grippers are shown as red and green circles having a dashed line representing their angles. From the top-left, the initial and final shapes, the error of the controlled points, and the gripper velocities on different axes can be seen. In the top-left image, the target points can be seen as a red diamond (as the target for the lower controlled point (shown in light blue)) and a blue diamond (as the target for the upper controlled point (shown in brown)). Subscript 1 is used for the lower controlled point with the red target and the lower gripper, and subscript 2 is used for the upper controlled point with the blue target and upper gripper.

Next, we take another pace and try to control the position of two and three non-adjacent points with the presented method using a single gripper. It is evident that the system becomes underactuated since the number of actuated DOF (three DOF) is lower than the number of DOF of the controlled points (four DOF for controlling two points and six DOF for controlling three points). Therefore, global stability cannot be guaranteed. Hence, the system can only be stable in a local sense. Different tests have been done, and the results can be seen in Fig. 3.14 to Fig. 3.19. The final shape of the object and the errors of the controlled points concerning their targets are presented in Fig. 3.14 to Fig. 3.19. As one can see, the errors converge to zero, and the presented method achieves its objective. The initial and target states of the controlled points in all tests are presented in Table 3.2. The final errors of all tests are also shown in that table.

3.6.5 Mean squared error (MSE)

To show the performance of the strategy, we will provide the mean squared error of all tests in this section. First, we define MSE to measure the mean squared errors between the target and final values in tests performed to control a point. Consider the fact that all tests' errors are within the same range. Consequently, we present MSE as:

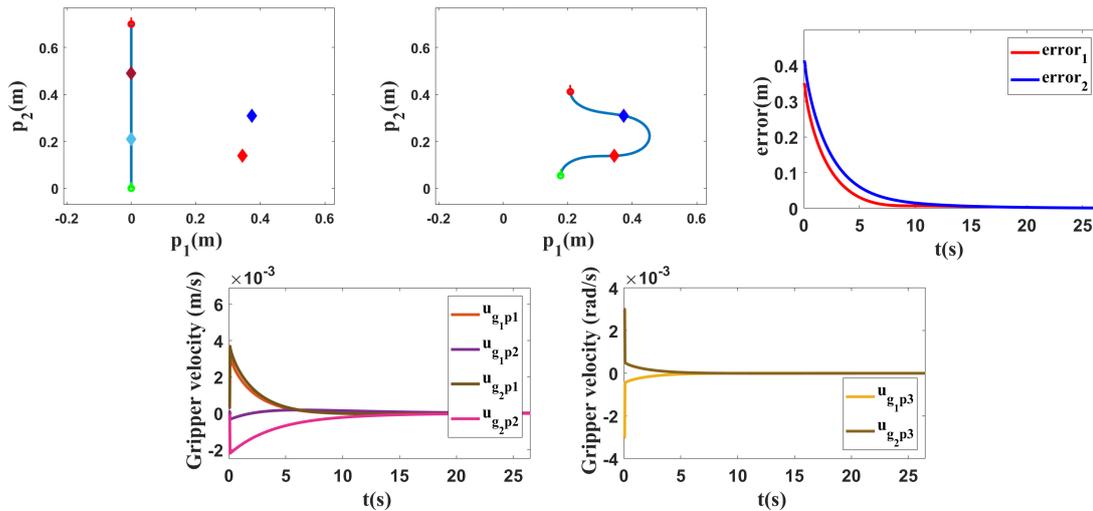


FIGURE 3.13: Simulation 3.5. Position control of two points using two grippers. The object is modeled using **KER** ($L = 0.70(m)$). From the top-left, the initial and final shapes, the error of the controlled points, and the gripper velocities on different axes can be seen. In the top-left image, the target points can be seen as a red diamond (as the target for the lower controlled point (shown in light blue)) and a blue diamond (as the target for the upper controlled point (shown in brown)). Subscript 1 is used for the lower controlled point with the red target and the lower gripper, and subscript 2 is used for the upper controlled point with the blue target and upper gripper.

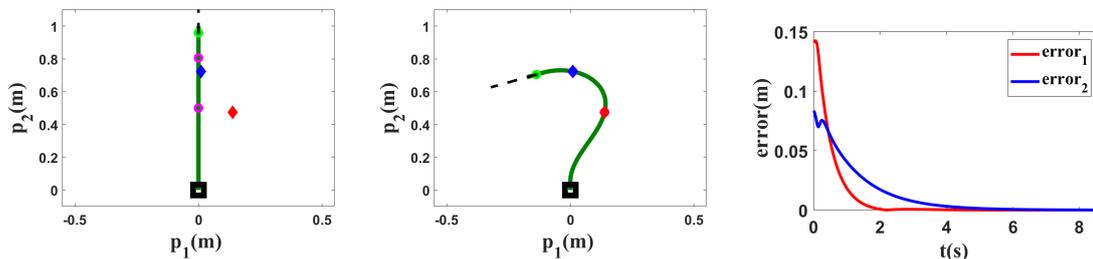


FIGURE 3.14: Simulation 3.6. Position control of two points using a gripper. The object is modeled using **ARAP** ($L = 1.00(m)$). From the left, the initial shape, final shape, and the errors of the three controlled points can be seen. The target points can be found as red (for the first controlled point which is in the lower section of the **LDO**) and blue (for the second controlled point which is in the upper section of the **LDO**) diamonds.

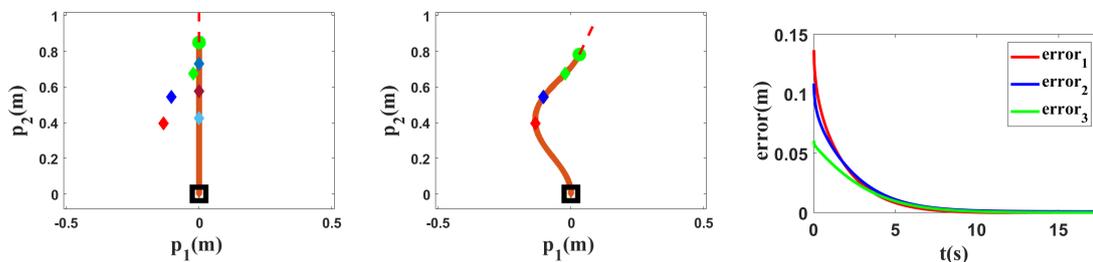


FIGURE 3.15: Simulation 3.7. Position control of three points using a gripper. The object is modeled using **ARAP** ($L = 0.85(m)$). From the left, the initial shape, final shape, and the errors of the three controlled points can be seen. The target point can be found as red (for the first control point that is lower), blue (for the second control point), and green (for the third control point that is upper) diamonds.

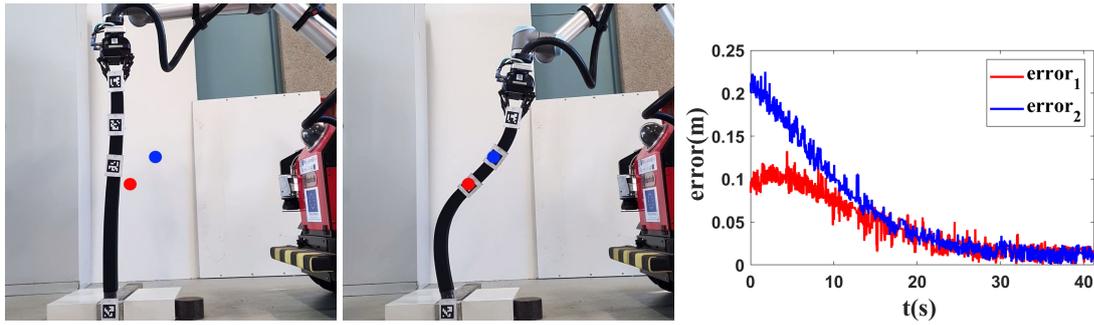


FIGURE 3.16: Experiment 3.4. Position control of two points. From the left, the initial shape, final shape of the object, and the errors of the three controlled points are indicated.

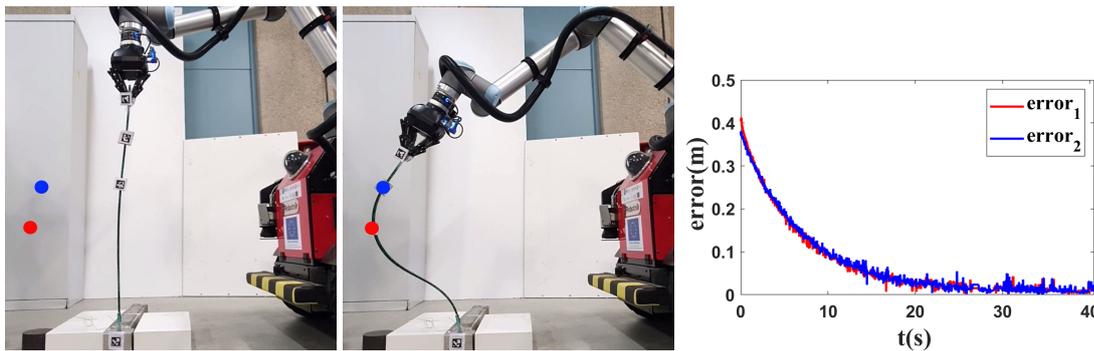


FIGURE 3.17: Experiment 3.5. Position control of two points. From the left, the initial shape, final shape of the object, and the errors of the three controlled points are indicated.

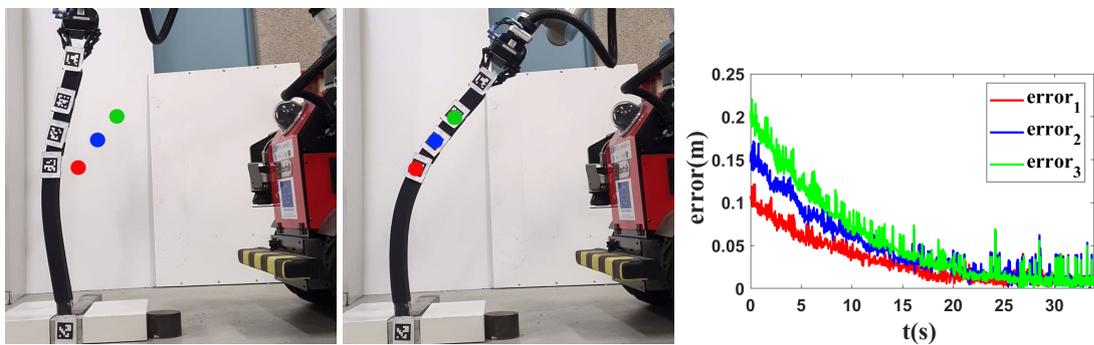


FIGURE 3.18: Experiment 3.6. Position control of three points. From the left, the initial shape, final shape of the object, and the errors of the three controlled points are indicated.

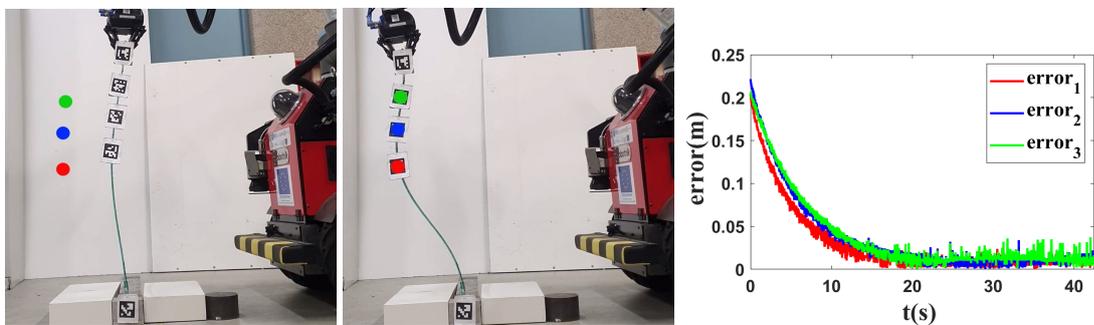


FIGURE 3.19: Experiment 3.7. Position control of three points. From the left, the initial shape, final shape of the object, and the errors of the three controlled points are indicated.

TABLE 3.2: State of the controlled points in the performed tests for controlling multiple points.

Test number	Initial position (m, m)	Target position (m, m)	Final error (m, m)
Sim. 3.4	[0.000, 0.210] [0.000, 0.490]	[-0.134, 0.397] [-0.104, 0.545]	[0.001, 0.000] [0.000, 0.001]
Sim. 3.5	[0.000, 0.210] [0.000, 0.490]	[0.344, 0.139] [0.373, 0.309]	[0.000, 0.001] [0.001, 0.001]
Sim. 3.6	[0.000, 0.500] [0.000, 0.800]	[0.139, 0.475] [0.010, 0.723]	[0.000, 0.000] [0.000, 0.000]
Sim. 3.7	[0.000, 0.425] [0.000, 0.578] [0.000, 0.731]	[-0.134, 0.397] [-0.104, 0.545] [-0.022, 0.675]	[0.000, 0.000] [0.001, 0.000] [0.000, 0.000]
Exp. 3.4	[0.047, 0.601] [0.072, 0.772]	[0.096, 0.505] [0.226, 0.619]	[0.009, 0.011] [0.005, 0.012]
Exp. 3.5	[0.037, 0.564] [0.072, 0.735]	[-0.345, 0.425] [-0.273, 0.584]	[0.009, 0.004] [0.004, 0.006]
Exp. 3.6	[-0.065, 0.529] [-0.044, 0.640] [-0.012, 0.753]	[0.040, 0.519] [0.101, 0.614] [0.177, 0.715]	[0.006, 0.003] [0.011, 0.006] [0.000, 0.006]
Exp. 3.7	[-0.044, 0.529] [-0.024, 0.659] [-0.006, 0.787]	[-0.232, 0.470] [-0.234, 0.601] [-0.204, 0.719]	[0.003, 0.007] [0.011, 0.002] [0.002, 0.004]

$$e_v = \sqrt{\frac{1}{n} \sum_{r=1}^n (\Delta v_r)^2} \quad (3.18)$$

where

$$\Delta v_r = |v_{rmf} - v_{rm-t}| \quad (3.19)$$

n is the number of tests, v_{rmf} is the final value of each state of the controlled point and v_{rm-t} is the target values of them in test r . Therefore based on the outlined results in section 3.6.3, for the simulation tests ($n = 3$) of controlling one point, we have:

$$e_{p_1} = 0.000(m), e_{p_2} = 0.000(m), e_{p_3} = 0.001(rad) \quad (3.20)$$

The experimental tests of section 3.6.3 ($n = 3$) are subjected to the same analysis, and the results are as follows:

$$e_{p_1} = 0.002(m), e_{p_2} = 0.003(m), e_{p_3} = 0.012(rad) \quad (3.21)$$

The same analysis are applied to the results of section 3.6.4 (controlling multiple points). However, we use the mean of the error of the controlled points in each test to obtain the Δv_r . For the simulation results ($n = 4$) we have:

$$e_{p_1} = 0.000(m), e_{p_2} = 0.000(m) \quad (3.22)$$

The MSE of the experimental results presented in section 3.6.4 ($n = 4$) obtain as follows:

$$e_{p_1} = 0.006(m), e_{p_2} = 0.007(m) \quad (3.23)$$

As shown, the proposed approach is able to precisely control the position of one or a few points on a deformable rod despite the object's various properties. The results presented in Table 3.1 and Table 3.2 show that the presented method works very accurately in simulation regardless of the type of simulated model we adopt to validate the method. However, the reason that the algorithm works in simulations slightly better than in experiments is due to the detection method that we use. Some noises exist in our detection method, which is why we have some small vibrations in our experimental results.

3.7 Conclusion

In this starting work on DO manipulation, we have presented an approach to control one or several arbitrary points along the length of an object. The proposed approach allows deforming elastic linear objects in a controlled way and needs only a simple offline geometric model, which is an important practical advantage. The limitations of the approach include: it is restricted to 2-D workspaces, it cannot handle non-smooth unstable deformation behaviors, the type of objects it can be applied on (fixed-length elastic LDOs) is limited. The method cannot be applied to the objects without the possibility to calculate the Jacobian matrix. This is a challenge since obtaining an offline Jacobian matrix has some limitations, as discussed in section 3.4.2. Another issue with this method is that the exact pose of the gripper must be known during the manipulation. Therefore, with the lesson learned from this starting research, in the next chapter, we present a method that can control the pose of the controlled point without the need to obtain a model-based Jacobian matrix or use any geometric model.

Chapter 4

Adaptive deformation control of LDOs without perceiving the entire shape

Although the performance of the method presented in the previous chapter is acceptable, it can only be applied to entirely linear objects of fixed length with a predefined fixed number of nodes. In this chapter, we aim to provide a more widely applicable technique for controlling LDOs than the method presented in the previous chapter. We aim to present a method in which the positions of the other nodes are not required to be known in advance, and we only need to know the position of the controlled node. The main application we consider in this chapter is in the field of agriculture, but may have interests in other tasks such as human daily activities and industrial production. We specifically consider an elastic linear object where one of its endpoints is grasped by a robotic arm. To deal with the mentioned problem, we propose a model-free method to control the state of an arbitrary point (called controlled point) in section 4.4 that can be at any place along the object's length. Our approach allows the robot to manipulate the object without knowing any model parameters or offline information of the object's deformation. An adaptive control strategy is proposed to adjust the state of the controlled point toward the desired location and deform the object automatically. The control law is developed thanks to the adaptive estimation of the system parameters and its states. This method can track a desired manipulation trajectory to reach the target point, which leads to a smooth deformation without drastic changes. A Lyapunov-based argument is presented for the asymptotic convergence of the system that shows the process's stability and convergence to desired state values. To validate the controller, numerical simulations involving two different deformation models are conducted in section 4.5, and performances of the proposed algorithm are investigated through full-scale experiments in section 4.6.

4.1 Introduction

In the literature of DO manipulation, some works have been done by considering an accurate object model [JP03,TFH99,KGKG98], as mentioned in the previous chapters. On the contrary, several recent works do not need an exact DO model by using some approximations of the deformation behavior [NAL17,HSP18,ZNAPC21,COC+20]. [NAL14] used an online estimation method to control the shape of the objects. However, the performance of the method presented by [NAL14] depends on the choice of regression matrix. A vision-based method that computes parameterized regression features to reshape an elastic rod is presented in [QMNA+20]. Similarly, [LKM20] proposed a model-free visual servoing approach for deforming a wire into a desired shape. These algorithms are used for low-speed movements and they need prior knowledge about the the deformation of

the object or to perceive the entire shape of the object during the manipulation to calculate the used deformation Jacobian. However, these models cannot be used for large deformations. Moreover, they mostly need offline information about the object to calculate a Jacobian matrix or to perceive the total shape of the object during the procedure. These methods cannot control the trajectory of the manipulation of the DOs since they were designed to reach fixed set-points.

This chapter focuses on an elastic linear rod manipulation with one of its ends fixed, with agriculture as the main targeted field of application. Many agriculture tasks need to reshape objects [Dav06, HS18, ZGT⁺18]. Most agriculture elements (such as branches) can be assumed as LDOs. The problem is that there are few works about the manipulation of DOs in this domain, and the existing DO manipulation approaches did not include this field in their research. Accordingly, this chapter aims to develop a method to be used in the agricultural field for some tasks such as pruning the trees and plant inspection.

As mentioned, most of the proposed methods in the literature need some information about the object (an exact model or an estimation of the deformation). They mostly need to perceive the total shape of the object (or enough points on the object to cover its entire shape) during the deformation process. Therefore, applying these methods with these limitations to the objects that exist in the agriculture field is not easy. This inspired us to develop a method to solve these issues and deform the object while moving a point on the object to a specific target. Therefore, we propose an adaptive control strategy, using a gripper grasping the object at a given point to regulate the state (position and orientation) of an arbitrary point on the body of the object. The proposed method does not need prior knowledge of any model parameters of the object and works in real-time. Our method does not need to perceive the entire shape of the object and does not attach to any particular model or object.

4.2 Our contributions

There are several contributions in our work:

- We take as starting point an adaptive control scheme proposed for biomedical applications [ASTZ17, ASTZ18] and we extend it to a novel problem which is DO manipulation.
- To the best of our knowledge, the existing adaptive approaches [NAL17, ZNF⁺18, NALRL14, LKM20] simply reach a fixed set-point without considering any deformation trajectory while our controller can be used to track a desired dynamic evolution of the state. This helps to have a closer control on the manipulation of the object and time of the task's completion. Furthermore, tracking these trajectories instead of a fixed point can be helpful in reaching large deformations.
- This method does not need to calculate a Jacobian based on a model to obtain the relationship between the deformation of the object and the motion of the gripper. The presented controller also does not need offline information of the object's deformation.
- Compared with [LWL19, NALRL14], the current method can control the full states of the controlled point, including the angle.
- Compared with [NAL17, ZNF⁺18], our work contains a formal analysis of the system dynamics under the controller.

Even though our method does not control the full shape of the object, controlling as we do the position and orientation at one point is sufficient for some tasks, requires to perceive online only a single point (not the full shape of the object), and implies the system is fully actuated, which can reduce the impact of local minima. One of the most interesting properties of our method is that it can be successfully applied on objects with fairly diverse characteristics. To demonstrate this, we provide an extensive validation including

simulation tests with different deformation models (KER, ARAP), and real robotic experiments with varied objects: a sponge rod, a plexiglass rod, and real vegetable plants¹.

4.3 Problem definition

This section defines the challenge we deal with in this chapter. We consider elastic linear objects, illustrated schematically in Fig. 4.1. The assumptions of our work are as follows. We assume that the object lies in a 2-D workspace, which is still a complex and interesting challenge [KFB⁺21, ZNF⁺18]. The object is represented as a continuous curve and its state by the position and orientation (tangent angle of the curve) at each point along its length. One of the ends is fixed, and the rest of the object is free to move. The length of the rod is denoted as L . We assume that the object deforms elastically and its size fits the workspace of the robotic arm. A robotic gripper is used to grasp rigidly the other end (which we call the grasped point g) on the object, and it can set the object's state at that point. Moreover, we assume that the shape of the object always stays in quasi-static stable equilibrium, and as shown in Fig. 4.1, the whole shape of the object can be manipulated by moving g . It should also be noted that the thickness range of the object is dependent on the opening-closing range of the gripper. A camera is used to track the pose of the controlled point (m), and it is assumed that the position and orientation of m can be known during the deformation process. The exact position and orientation of the target point are known before the start of the procedure, and they are defined in the reachable range of the robot workspace where the object can be reached (i.e., the manipulation task is feasible).

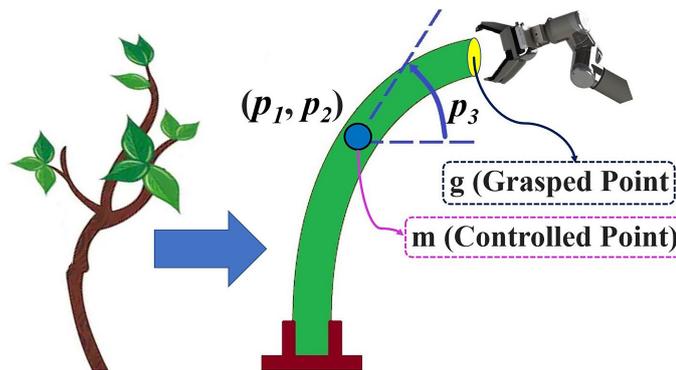


FIGURE 4.1: Considered object in the current study.

The problem we address is the control of the state of an arbitrarily chosen point (m) on the body of the object using g , and the manipulation task is to move m to the target position and orientation M_f . M denotes the state of the controlled point m , and G denotes the state of the grasped point g , which are defined as:

$$M = [p_{1m}, p_{2m}, p_{3m}]^T, \quad G = [p_{1g}, p_{2g}, p_{3g}]^T \quad (4.1)$$

The main difficulty is that solving this indirect control problem requires taking into account the deformation behavior of the object appropriately. To solve the problem, we will design an indirect adaptive controller, to be discussed later in this chapter.

¹A video of the simulations and experimental results can be found at: <https://www.youtube.com/watch?v=mMsRmehdcXo>

4.4 Controller design

In this section, an adaptive control method to manipulate a point along the object's length by moving the gripper is described. Our objective is to develop a controller to reshape the object so that m reaches its target configuration without the need to know any model parameters. The final state of the target point is denoted by $M_f = [p_{1mf}, p_{2mf}, p_{3mf}]^T$. Like [ZNF⁺18, LKM20, NAL17, HHS⁺19, YZZL21], the control input is specified as:

$$U = \dot{G} \quad (4.2)$$

To control the LDOs properly, we try to set an appropriate relationship between the motion of the gripper and the controlled point. In this context, we were inspired by [Ogd97, NALRL14] which explained that a relationship exists between G and M :

$$M = f_1(\dot{G}) \quad \text{or} \quad \dot{M} = f_2(\dot{G}) \quad (4.3)$$

Accordingly, for the motion of g , one can write:

$$\dot{G} = F(M, \dot{M}) \quad (4.4)$$

In the current study, the objective is to design a control law for $U = [u_1, u_2, u_3]^T$. However, we emphasize that the mentioned existing works can only control the instantaneous variation of the state of the controlled variables. By doing so, they can make the state reach a fixed set-point. In this work, we introduce additional terms in the structure of the controller. These terms allow us to track a desired trajectory of the state, instead of just reaching a fixed set-point. The suitability of the controller formulation that we propose is supported by the fact that it follows well-known adaptive control techniques [SL⁺91] and previous works in other application contexts [ASTZ17, ASTZ18]. Hence, in this study, the relationship between the grasped point and the controlled point is assumed as follows:

$$\dot{G} = \lambda \dot{M} + \Omega \tilde{M} \quad (4.5)$$

where Ω is a 3×3 matrix that consists of some unknown parameters we named a_{ij} (i & $j = 1, 2, 3$) and \tilde{M} is $M - M_{des}$. $M_{des}(t)$ is the desired trajectory of M . This trajectory is defined as bounded and differentiable. Our goal is to design an adaptive indirect controller to track this trajectory. Note that we use the desired state variable M_{des} in the dynamic equation (4.5) to reach the target point in a limited time based on a desired deformation trajectory. λ is then defined as a vector with positive components $[a_{10}, a_{20}, a_{30}]$, which permits to account for the speed of the controlled point m during the process. Therefore, according to (4.5), one can write the following relationship between the motion of m and g :

$$\dot{p}_{ig} = a_{i0} \dot{p}_{im} + \sum_{j=1}^3 a_{ij} \tilde{p}_{jm} \quad (4.6)$$

where the tracking errors \tilde{p}_{im} are $p_{im} - p_{imdes}$. We introduce p_{imdes} as the desired state values of the controlled point m , which are time-varying to track a desired trajectory of the state by the adaptive controller. They will be defined later. a_{ik} ($k = 0, j$) are dependent on the current shape of the object, since they relate the gripper to the control point. It is reasonable to assume these parameters are approximately constant in a local neighborhood of the current operating point (i.e., current shape) since they change smoothly. As a result, to build a control based on this model, these parameters (a_{ik}) have to be estimated and online updated. An adaptive estimation algorithm is here introduced to estimate these parameters.

4.4.1 Adaptive estimation algorithm

In this section, the adaptive control strategy is developed for (4.6). Fig. 4.2 shows the block diagram of the proposed controller.

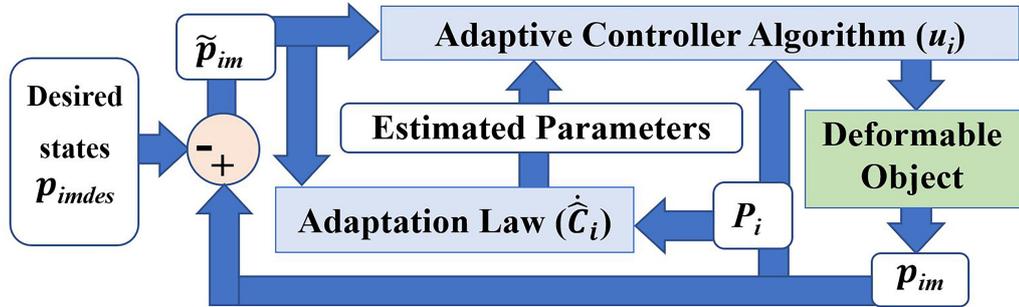


FIGURE 4.2: Schematic diagram of the proposed adaptive control strategy.

First, we rearrange the right-hand side of (4.6).

$$a_{i0}\dot{p}_{im} + \sum_{j=1}^3 a_{ij}\tilde{p}_{jm} = R_i C_i^T \quad (4.7)$$

where:

$$R_i = [\dot{p}_{im}, \tilde{p}_{1m}, \tilde{p}_{2m}, \tilde{p}_{3m}] \quad (4.8)$$

$$C_i = [a_{i0}, a_{i1}, a_{i2}, a_{i3}] \quad (4.9)$$

With the purpose of finding the estimation of a_{ik} , the adaptive control law ($[u_1, u_2, u_3]^T$) is defined as:

$$u_i = \hat{a}_{i0}(\dot{p}_{imdes} - \alpha_i \tilde{p}_{im}) + \sum_{j=1}^3 \hat{a}_{ij} \tilde{p}_{jm} \quad (4.10)$$

The goal of this controller is to set the speed of m with the proposed forms $(\dot{p}_{imdes} - \alpha_i \tilde{p}_{im})$, which will help us to generate desired trajectories for the velocities to be used in the controller design. α_i are some positive constants. The accent $\hat{\cdot}$ is used for the estimated parameters of the system. In other words, \hat{a}_{ik} are some estimates of the time-varying parameters a_{ik} that will be updated using an adaptation law. They are the estimations of the true parameters a_{ik} .

By rearranging (4.10), the control law can be expressed as:

$$u_i = P_i \hat{C}_i^T \quad (4.11)$$

where:

$$P_i = [(\dot{p}_{imdes} - \alpha_i \tilde{p}_{im}), \tilde{p}_{1m}, \tilde{p}_{2m}, \tilde{p}_{3m}] \quad (4.12)$$

The vectors of system parameters' estimations are:

$$\hat{C}_i = [\hat{a}_{i0}, \hat{a}_{i1}, \hat{a}_{i2}, \hat{a}_{i3}] \quad (4.13)$$

The adaptation law for updating the estimated parameters is defined as:

$$\dot{\hat{C}}_i = -P_i T_i \tilde{p}_{im} \quad (4.14)$$

The adaptation gains T_i are constant non-singular positive definite matrices of size 4×4 . Note that, interestingly, the information that needs to be measured to implement the proposed adaptive feedback controller is just the value of M at each instant of time. The controller will be analyzed in the next section using the Lyapunov formalism.

Proposition: Under the action of the proposed control law u_i (4.10), the control system tracks the desired trajectory, i.e., $\tilde{p}_{im} \rightarrow 0$ as $t \rightarrow \infty$ for $i = 1, 2, 3$.

Proof: The closed-loop dynamics of the system using the proposed controller are obtained in this part. For this purpose, the control law (4.10) is substituted in (4.6), and by adding and subtracting a term, we have:

$$a_{i0} \dot{p}_{im} + \sum_{j=1}^3 a_{ij} \tilde{p}_{jm} = \hat{a}_{i0} (\dot{p}_{imdes} - \alpha_i \tilde{p}_{im}) + \sum_{j=1}^3 \hat{a}_{ij} \tilde{p}_{jm} + a_{i0} (\dot{p}_{imdes} - \alpha_i \tilde{p}_{im}) - a_{i0} (\dot{p}_{imdes} - \alpha_i \tilde{p}_{im}) \quad (4.15)$$

By rearranging (4.15), we have:

$$a_{i0} (\dot{p}_{im} - (\dot{p}_{imdes} - \alpha_i \tilde{p}_{im})) = (\dot{p}_{imdes} - \alpha_i \tilde{p}_{im}) (\hat{a}_{i0} - a_{i0}) + \sum_{j=1}^3 \tilde{p}_{jm} (\hat{a}_{ij} - a_{ij}) \quad (4.16)$$

According to (4.12), (4.13) and (4.16), the system's closed-loop dynamics for every i ($i = 1, 2, 3$) is found as:

$$a_{i0} (\dot{\tilde{p}}_{im} + \alpha_i \tilde{p}_{im}) = P_i \tilde{C}_i^T \quad (4.17)$$

note that $\tilde{C}_i = \hat{C}_i - C_i$. By rearranging (4.17), the dynamics are reformulated as:

$$\dot{\tilde{p}}_{im} = -\alpha_i \tilde{p}_{im} + \frac{1}{a_{i0}} P_i \tilde{C}_i^T \quad (4.18)$$

A positive definite Lyapunov function candidate is used next to analyze the system stability and the tracking convergence using the proposed controller.

$$V = \frac{1}{2} \left(\sum_{i=1}^3 \tilde{p}_{im}^2 + \frac{1}{a_{i0}} \tilde{C}_i T_i^{-1} \tilde{C}_i^T \right) \geq 0 \quad (4.19)$$

The time derivative of V is obtained as:

$$\dot{V} = \sum_{i=1}^3 (\tilde{p}_{im} \dot{\tilde{p}}_{im} + \frac{1}{a_{i0}} \dot{\tilde{C}}_i T_i^{-1} \tilde{C}_i^T) \quad (4.20)$$

We can write $\dot{\tilde{C}}_i = \dot{\hat{C}}_i$ since \dot{C}_i is negligible compared to $\dot{\hat{C}}_i$. Consequently, via the closed-loop dynamics (4.18), (4.20) can be expressed as:

$$\dot{V} = \sum_{i=1}^3 \left(-\alpha_i \tilde{p}_{im}^2 + \frac{1}{a_{i0}} P_i \tilde{C}_i^T \tilde{p}_{im} + \frac{1}{a_{i0}} \dot{\hat{C}}_i T_i^{-1} \tilde{C}_i^T \right) \quad (4.21)$$

Using the parameters' adaptation law (4.14), the time derivative of Lyapunov function (4.21) is simplified to:

$$\dot{V} = \sum_{i=1}^3 -\alpha_i \tilde{p}_{im}^2 \quad (4.22)$$

As one can see, the time derivative of the Lyapunov function is negative semi-definite.

The Lyapunov function proposed in (4.19) is positive definite ($V(t) > 0$) in terms of \tilde{p}_{im} and \tilde{C}_i . Its time derivative is negative semi-definite ($\dot{V} \leq 0$); therefore, $V(t)$ is bounded. Accordingly, one can conclude that \tilde{p}_{im} and \tilde{C}_i remain bounded. In addition, the desired trajectories of the states (p_{imdes}) are defined bounded; thus, the boundedness of $p_{im} = (p_{imdes} + \tilde{p}_{im})$ is also concluded. Now, the function $b(t)$ is defined:

$$b(t) = \sum_{i=1}^3 \alpha_i \tilde{p}_{im}^2 \geq 0 \quad (4.23)$$

such that $b(t) = -\dot{V}(t) \geq 0$, whose integration gives:

$$V(0) - V(\infty) = \lim_{t \rightarrow \infty} \int_0^t b(\eta) d\eta \quad (4.24)$$

Considering that $\dot{V} = dV/dt$ is negative semi-definite, $(V(0) - V(\infty))$ is positive and finite. Therefore, $\lim_{t \rightarrow \infty} \int_0^t b(\eta) d\eta$ has a finite positive value. Moreover, according to the Barbalat's lemma [SL+91], if $b(t)$ is uniformly continuous and the limit of integral $\lim_{t \rightarrow \infty} \int_0^t b(\eta) d\eta$ exists, then:

$$\lim_{t \rightarrow \infty} b(t) = 0 \quad (4.25)$$

Based on (4.23) and considering (4.25), it is concluded that:

$$\lim_{t \rightarrow \infty} \left(\sum_{i=1}^3 \alpha_i \tilde{p}_{im}^2 \right) = 0 \quad (4.26)$$

It is known that $\alpha_i > 0$ are non-zero positive constants and $\tilde{p}_{im}^2 \geq 0$ are positive. As a result, (4.26) implies that $\tilde{p}_{im} \rightarrow 0$ as $t \rightarrow \infty$. Consequently, the states of the controlled points converge to their corresponding desired values ($p_{im} \rightarrow p_{imdes}$).

4.5 Simulation results

To evaluate the proposed adaptive control strategy, the controller (4.11) and the adaption law for updating the estimated parameters (4.14) are simulated. At each time instant (t), using (4.11) the position and orientation of the grasped point are updated, and the new shape of the object is obtained using the simulated deformation models. It is assumed that the rod's shape remains stable during the entire control process. Intending to obtain a smooth and uniform dynamic evolution of the object's shape, we define exponential desired evolutions of the state. These allow reaching the desired final state of the controlled point in a limited time. Specifically, the following desired states are intended to be tracked using the proposed control strategy:

$$p_{imdes} = (p_{im0} - p_{imf}) \exp(-g_i t) + p_{imf} \quad (4.27)$$

p_{im0} are the initial values of M . The parameters g_i (the exponential coefficient) are positive constants.

4.5.1 Simulations using Kirchhoff elastic rod model

In the first step of this part, we have chosen the KER formalism [BM14, SS16] to use in our tests. Various conditions have been used in each simulation to show the performance of the proposed methodology. L , initial position and final position of the rod have been

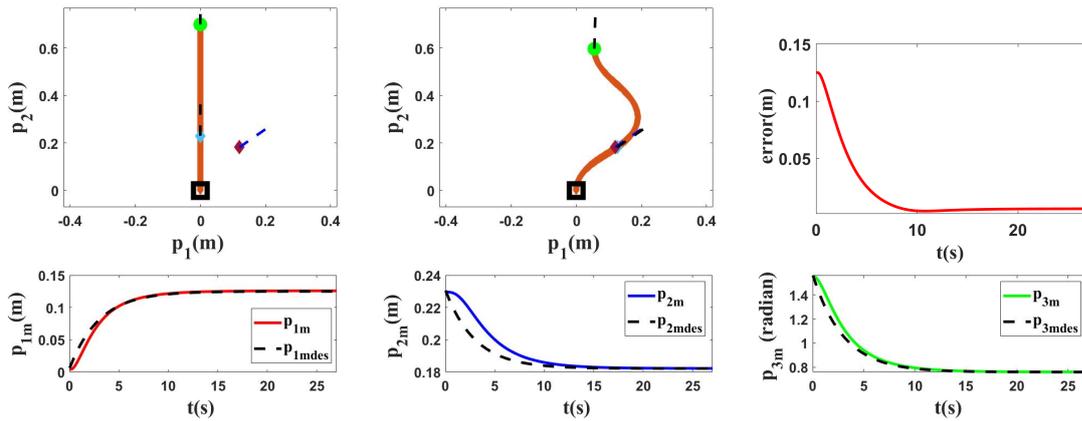


FIGURE 4.3: Simulation 4.1. From the top-left in the first row, the initial and final shape of the object and the servoing error of the controlled point are illustrated. The desired state of the point m is shown as a purple diamond (position) and a dashed line segment (orientation). L is $0.70(m)$. Second row shows the states of the controlled point regarding their desired trajectories in the first simulation using the KER model.

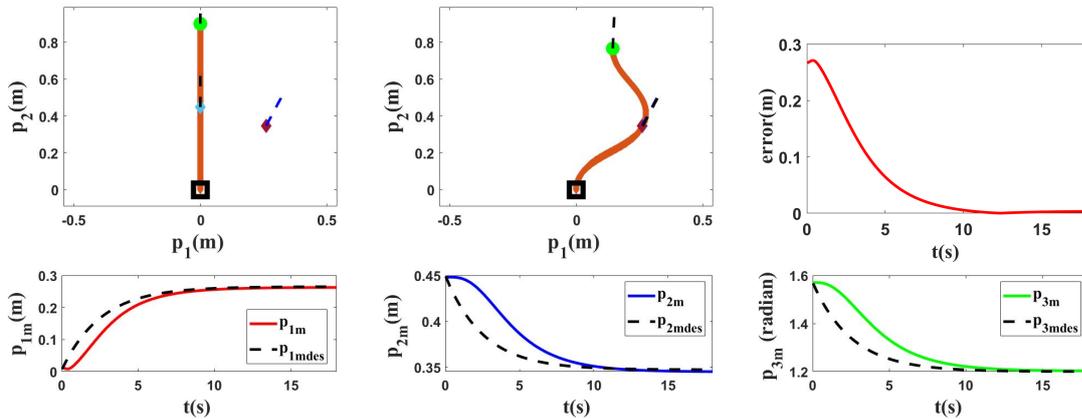


FIGURE 4.4: Simulation 4.2. From the top-left in the first row, the initial and final shape of the object and the servoing error of the controlled point are illustrated. The desired state of the point m is shown as a purple diamond (position) and a dashed line segment (orientation). L is $0.90(m)$. Second row shows the states of the controlled point regarding their desired trajectories in the second simulation using the KER model.

changed in each simulation, and one can find the results in Fig. 4.3 to Fig. 4.5. p_1 and p_2 are measured in meters and p_3 in radians. In the first simulation, L is set to $0.70(m)$, and the initial position and target of the controlled point are set as shown in Fig. 4.3. In the next simulations, we changed L to $0.90(m)$ to provide different tests with different conditions, and we applied the proposed method to control the position and orientation of two different controlled points with two different targets. The first row of Fig. 4.3 to Fig. 4.5 displays how the object changed from the initial shape to the final shape in these simulations. The second row of Fig. 4.3 to Fig. 4.5 shows that the proposed strategy can control the manipulated object according to the desired trajectories. The initial and target pose of these simulations are given in Table 4.1.

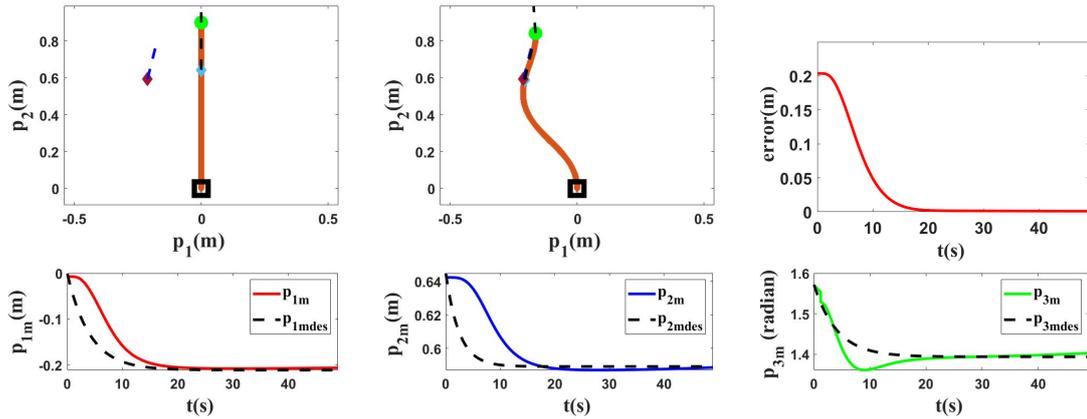


FIGURE 4.5: Simulation 4.3. From the top-left in the first row, the initial and final shape of the object and the servoing error of the controlled point are illustrated. The desired state of the point m is shown as a purple diamond (position) and a dashed line segment (orientation). L is $0.90(m)$. Second row shows the states of the controlled point regarding their desired trajectories in the third simulation using the **KER** model.

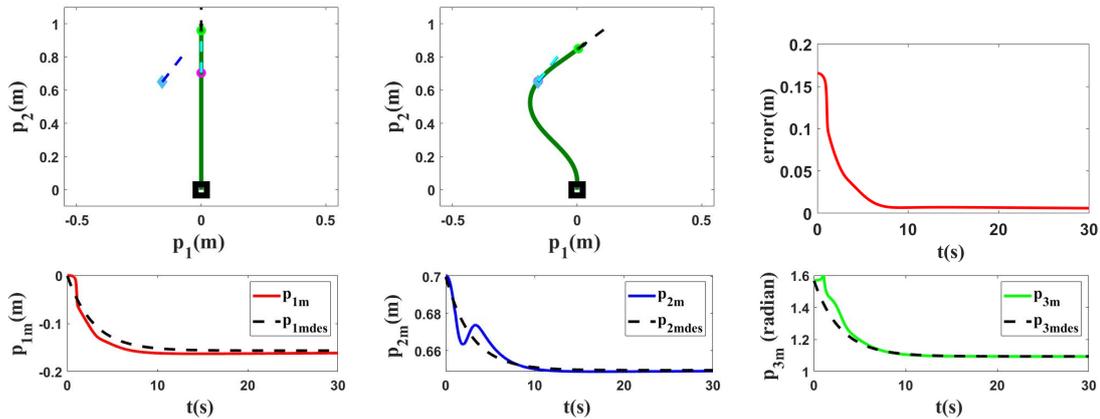


FIGURE 4.6: Simulation 4.4. From the top-left in the first row, the initial and final shape of the object and the servoing error of the controlled point are displayed. The desired state of the point m is shown as a blue diamond (position) and a dashed line segment (orientation). $L = 1.00(m)$. Second row depicts the states of the controlled point based on the desired trajectories in the fourth simulation using the **ARAP** model.

4.5.2 Simulations using ARAP model

In this part, we use another method to model the studied **DO**. These simulations aim to show that the proposed controller can be used with different models with different dynamic parameters. To do that, we use **ARAP** surface modeling [SA07] to model the object. To validate the performance of the proposed controller, three different simulations are conducted, and in each of them, various conditions are applied to the object which means L , M_0 , and M_{fdes} have been changed. One can find the results in Fig. 4.6 to Fig. 4.8. The first row of Fig. 4.6 to Fig. 4.8 displays the initial and final shape of the object in these simulations. The second row of Fig. 4.6 to Fig. 4.8 demonstrates the deformation of the states based on the desired trajectories. It shows that the proposed method can control the object following desired trajectories. The initial and target pose of these simulations are given in Table 4.1.

As shown by the various simulations under different conditions (various models, lengths, initial positions, and targets), it is clear that with the proposed method we can

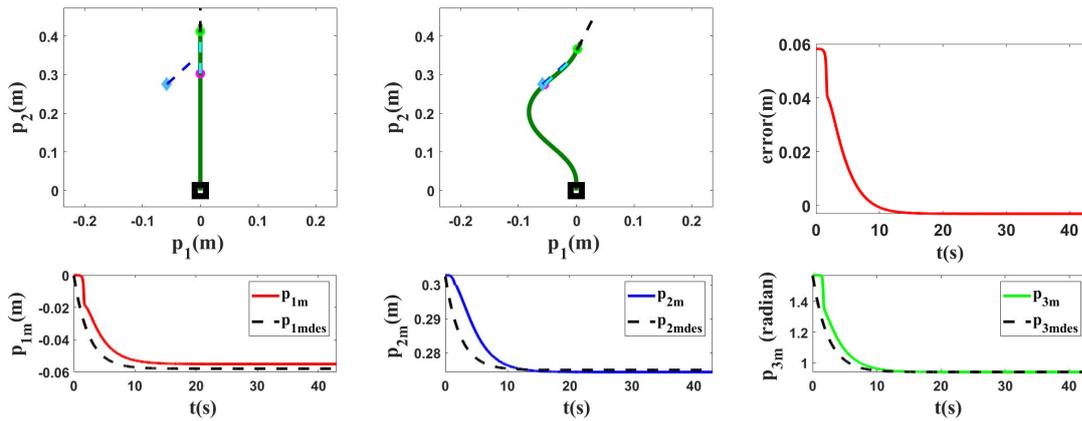


FIGURE 4.7: Simulation 4.5. From the top-left in the first row, the initial and final shape of the object and the servoing error of the controlled point are displayed. The desired state of the point m is shown as a blue diamond (position) and a dashed line segment (orientation). $L = 0.43(m)$. Second row depicts the states of the controlled point based on the desired trajectories in the fifth simulation using the ARAP model.

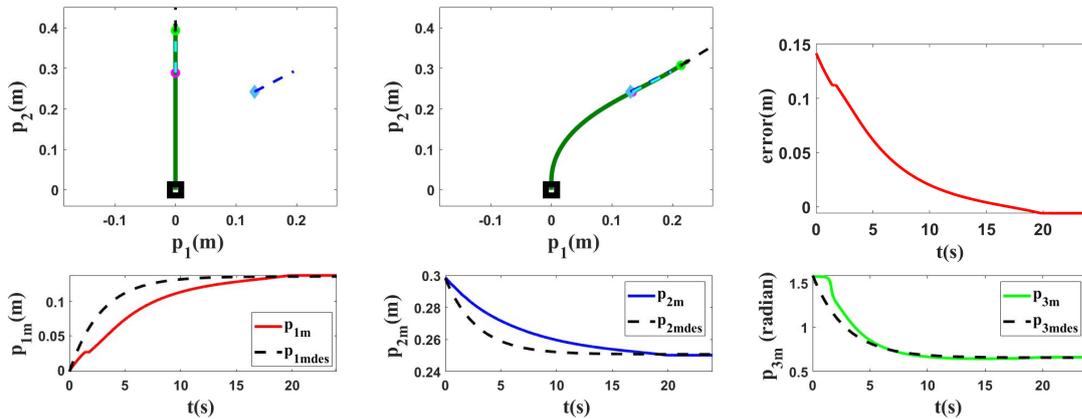


FIGURE 4.8: Simulation 4.6. From the top-left in the first row, the initial and final shape of the object and the servoing error of the controlled point are displayed. The desired state of the point m is shown as a blue diamond (position) and a dashed line segment (orientation). $L = 0.43(m)$. Second row depicts the states of the controlled point based on the desired trajectories in the sixth simulation using the ARAP model.

TABLE 4.1: State of the controlled points in the performed tests.

Simulation number	Initial pose (m, m, rad)	Target pose (m, m, rad)	Final error (m, m, rad)
Sim. 4.1	[0.0000, 0.2310, 1.5708]	[0.1200, 0.1820, 0.7600]	[0.0006, 0.000, 0.0010]
Sim. 4.2	[0.0000, 0.4500, 1.5708]	[0.2600, 0.3470, 1.2000]	[0.0000, 0.0003, 0.0000]
Sim. 4.3	[0.0000, 0.6430, 1.5708]	[-0.2110, 0.5896, 1.3948]	[0.0021, 0.0005, 0.0062]
Sim. 4.4	[0.0000, 0.7000, 1.5708]	[-0.1564, 0.6495, 1.0952]	[0.0001, 0.0041, 0.0000]
Sim. 4.5	[0.0000, 0.3000, 1.5708]	[-0.0579, 0.2750, 0.9406]	[0.0029, 0.0006, 0.0046]
Sim. 4.6	[0.0000, 0.3000, 1.5708]	[0.1400, 0.2500, 0.6650]	[0.0015, 0.0002, 0.0023]

move the point to the chosen goal with an acceptable error.

4.6 Robot experiments

In this section, we conduct different experiments to show the performance of the proposed adaptive control strategy on real objects. The object is rigidly grasped by the robotic gripper, and it is tracked with a fixed camera using Aruco markers. The object's shape is captured by a Logitech C270 camera using OpenCV. It should be noted that we only track the controlled point and obtain its states at each time step. The whole experimental setup is found in Fig. 4.9.

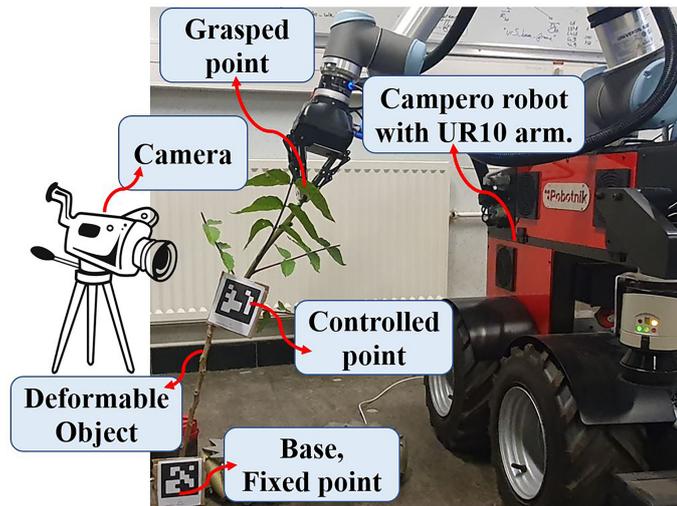


FIGURE 4.9: The experimental setup used in this work.

4.6.1 Experiments with a sponge rod

For the first and second experiments, we use a sponge rod ($L = 0.43(m)$) which has high flexibility. In this part, two different controlled points and two different targets are set to validate the proposed methodology. We use the same target in the first experiment as in the last simulation since the object's length is the same. With this test, we aim to show that we can expect almost the same results in simulations and experiments for similar conditions. The obtained results (Fig. 4.8 and Fig. 4.10) confirm that we obtain almost the same behavior in experiment and simulation. In the second experiment, we set another target and initial position for the controlled point to show the performance of the used method in different conditions. In the first row of Fig. 4.10 and Fig. 4.11, the initial and final configuration of the object can be seen. The second row of Fig. 4.10 and Fig. 4.11 shows the evolution of the states in these experiments.

4.6.2 Experiments with a plexiglass rod

For the third and fourth experiments, we use an LDO made of plexiglass ($L = 0.95(m)$). In this section, our objective is to show that the proposed method can work with different objects having different characteristics. The initial and final shapes of the object in these experiments are found in the first row of Fig. 4.12 and Fig. 4.13. The second row of Fig. 4.12 and Fig. 4.13 displays how the states are changed during these tests.

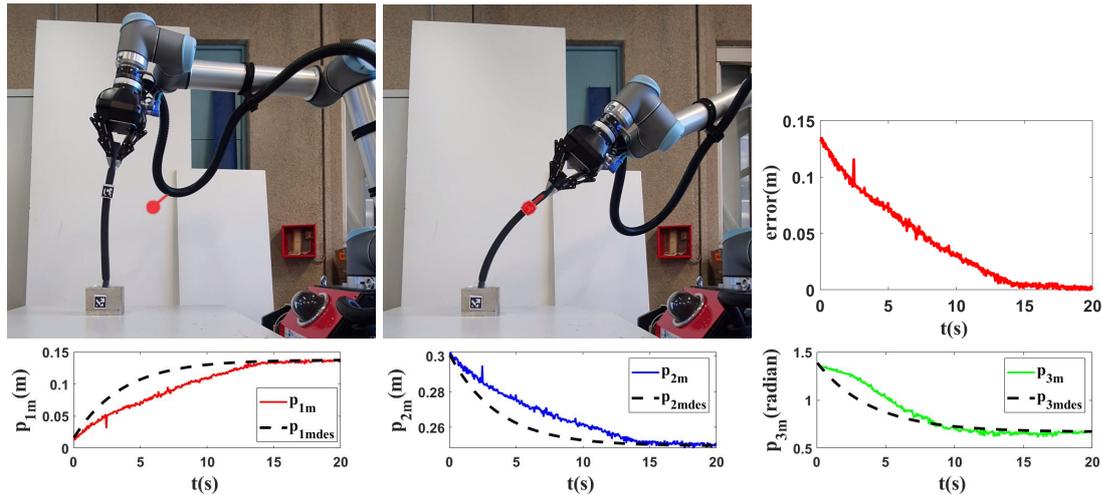


FIGURE 4.10: Experiment 4.1. From the top-left in the first row, the initial and final shape of the object and the servoing error of the controlled point are illustrated. The desired state of m is shown as a red circle with a small line segment. Second row presents the evolution of the states in the first experiment with respect to their desired trajectories.

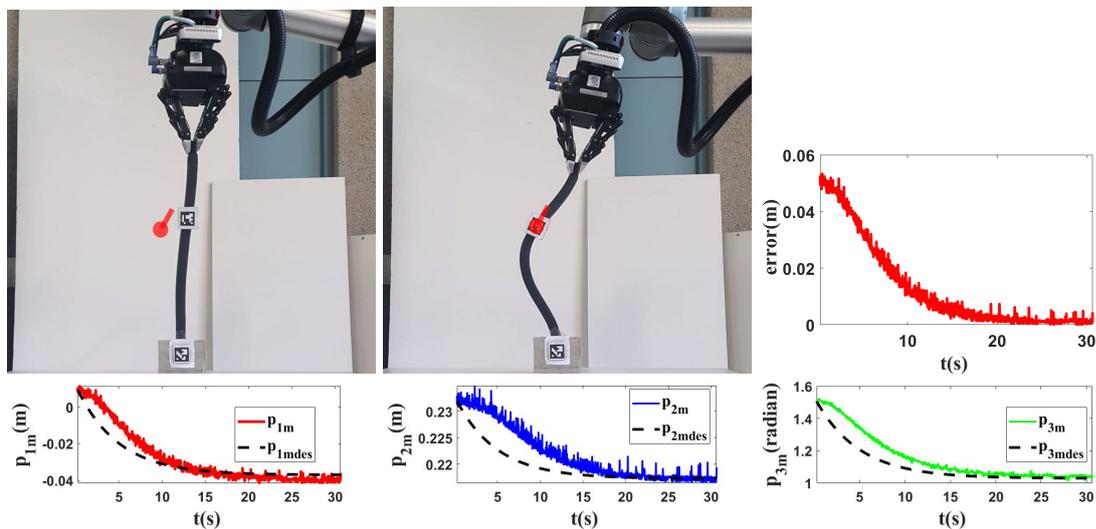


FIGURE 4.11: Experiment 4.2. From the top-left in the first row, the initial and final shape of the object and the servoing error of the controlled point are illustrated. The desired state of m is shown as a red circle with a small line segment. Second row presents the evolution of the states in the second experiment with respect to their desired trajectories.

4.6.3 Experiments with real vegetables

For the last experiments, we use two different real vegetables and we do the tests with them. Our goal is to show that the proposed method can be used with natural agricultural rods (vegetable branches) without knowing their properties. The initial and final shapes of the objects are presented in the first row of Fig. 4.14 to Fig. 4.16. Second row of Fig. 4.14 to Fig. 4.16 demonstrates the evolution of the states of the controlled points based on the desired trajectories in these experiments. As shown by the results (Fig. 4.14 to Fig. 4.16), with the proposed method, the controlled point on the object can reach its target with a tolerable error.

The aforementioned results (Fig. 4.10 to Fig. 4.16) show that the proposed strategy can control the object in a precise way, and with this method, one can manipulate the

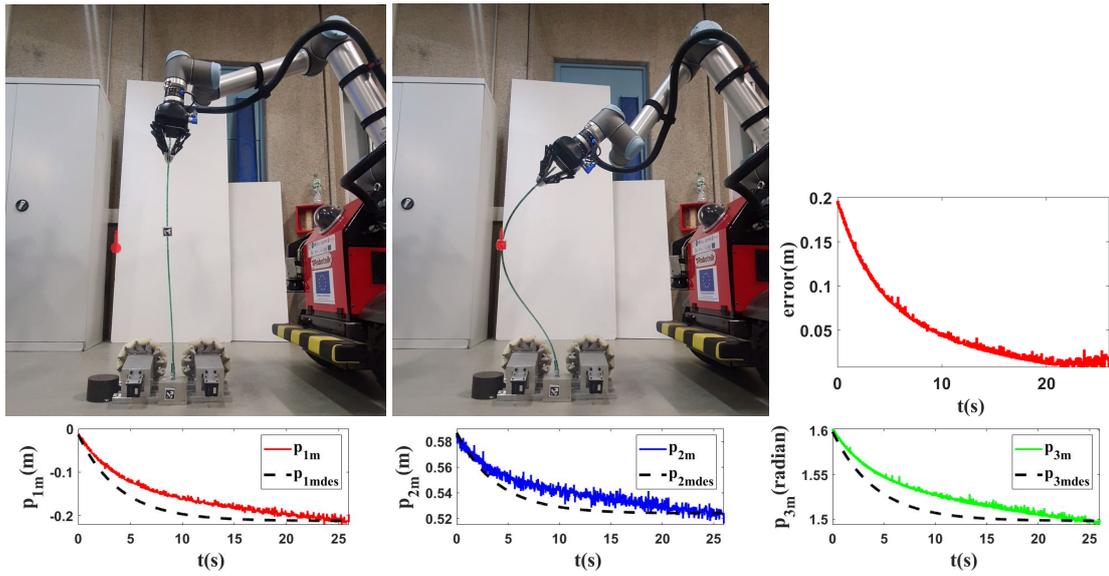


FIGURE 4.12: Experiment 4.3. From the top-left in the first row, the initial and final shape of the object and the servoing error of the controlled point are illustrated. The desired state of m is shown as a red circle with a small line segment. Second row presents the states of the controlled point in the third experiment with respect to their desired trajectories.

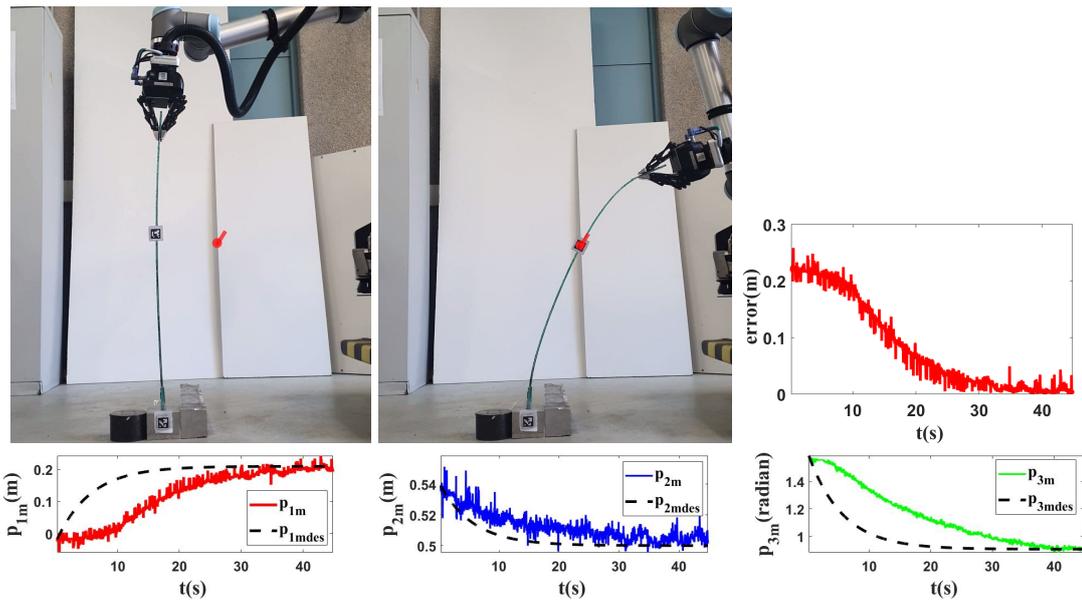


FIGURE 4.13: Experiment 4.4. From the top-left in the first row, the initial and final shape of the object and the servoing error of the controlled point are illustrated. The desired state of m is shown as a red circle with a small line segment. Second row presents the states of the controlled point in the fourth experiment with respect to their desired trajectories.

taken object desirably following the desired trajectories. The initial and target pose of all experiments are given in Table 4.2.

4.6.4 Mean squared error (MSE)

In this section, we will present the mean squared error of all tests to show the performance of the method. We defined MSE (3.18) as the average of the squares of the errors

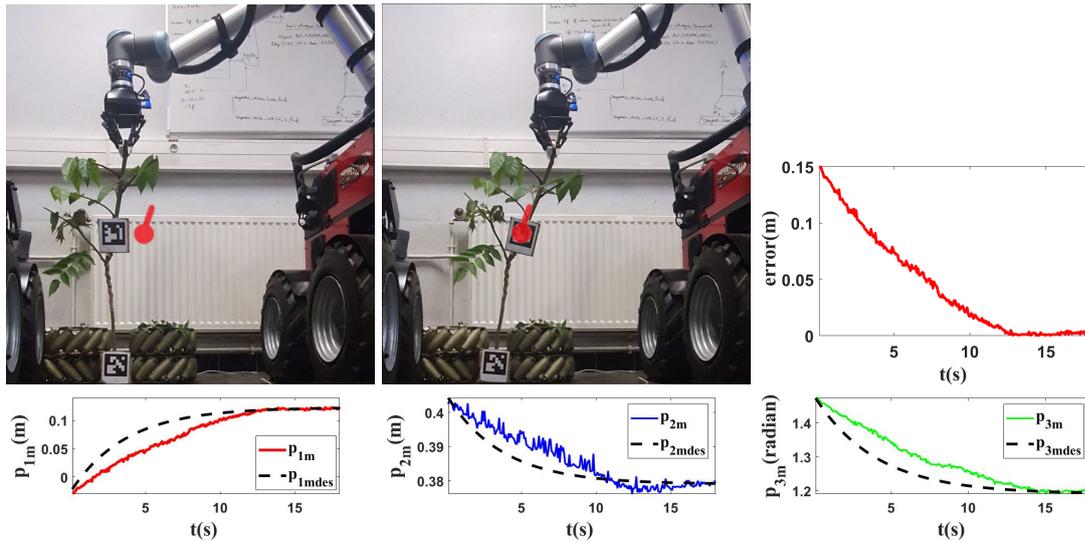


FIGURE 4.14: Experiment 4.5. From the top-left in the first row, the initial and final shape of the object and the servoing error of the controlled point are illustrated. The desired state of m is shown as a red circle with a small line segment. Object's length (L) is $0.72(m)$. Second row presents the evolution of the states in the fifth experiment with respect to their desired trajectories.

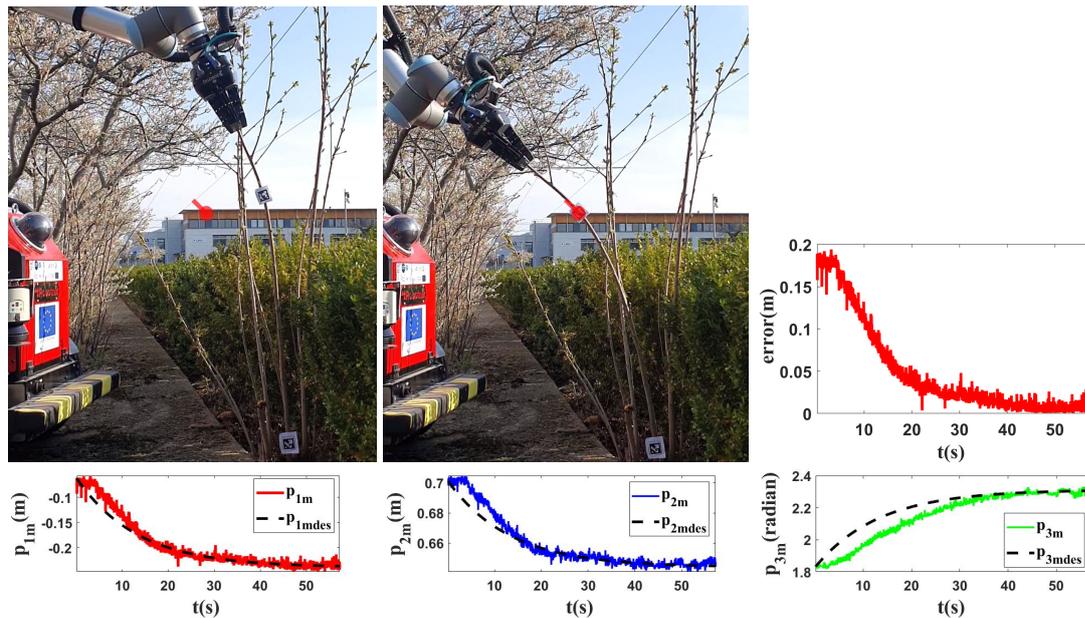


FIGURE 4.15: Experiment 4.6. From the top-left in the first row, the initial and final shape of the object and the servoing error of the controlled point are illustrated. The desired state of m is shown as a red circle with a small line segment. Object's length (L) is $0.99(m)$. Second row presents the evolution of the states in the last experiment with respect to their desired trajectories.

between the target values and the final values in the previous chapter. The errors of all tests are in the same range. Therefore, based on the presented results in the previous section, for the simulation tests ($n = 6$), we have:

$$e_{p_1} = 0.0016(m), e_{p_2} = 0.0017(m), e_{p_3} = 0.0033(rad) \quad (4.28)$$

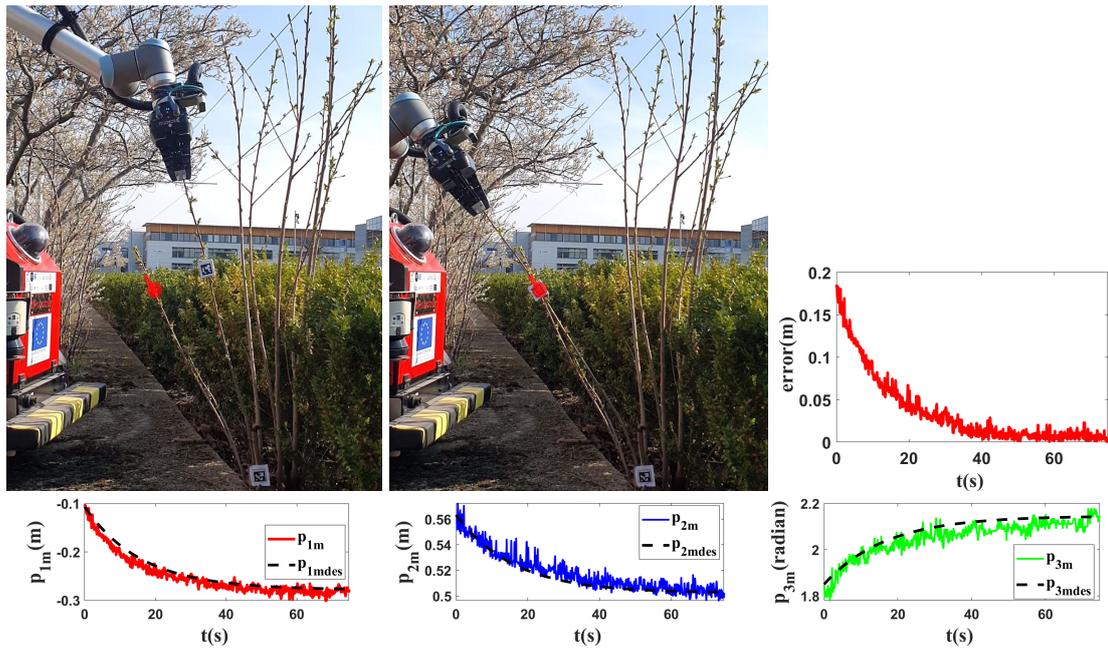


FIGURE 4.16: Experiment 4.7. From the top-left in the first row, the initial and final shape of the object and the servoing error of the controlled point are illustrated. The desired state of m is shown as a red circle with a small line segment. Object's length (L) is $0.86(m)$. Second row presents the evolution of the states in the last experiment with respect to their desired trajectories.

TABLE 4.2: State of the controlled points in the performed tests.

Experiment number	Initial pose (m, m, rad)	Target pose (m, m, rad)	Final error (m, m, rad)
Exp. 4.1	[0.016, 0.300, 1.428]	[0.140, 0.250, 0.665]	[0.003, 0.003, 0.004]
Exp. 4.2	[0.010, 0.232, 1.507]	[-0.036, 0.217, 1.030]	[0.003, 0.001, 0.013]
Exp. 4.3	[-0.016, 0.585, 1.608]	[-0.211, 0.523, 1.498]	[0.002, 0.004, 0.003]
Exp. 4.4	[-0.019, 0.539, 1.578]	[0.209, 0.500, 0.908]	[0.007, 0.003, 0.000]
Exp. 4.5	[-0.025, 0.405, 1.480]	[0.123, 0.380, 1.200]	[0.001, 0.003, 0.003]
Exp. 4.6	[-0.062, 0.702, 1.833]	[-0.237, 0.645, 2.306]	[0.004, 0.001, 0.003]
Exp. 4.7	[-0.106, 0.563, 1.849]	[-0.277, 0.503, 2.143]	[0.002, 0.006, 0.011]

The same analysis is carried out on the experimental tests ($n = 7$), and the results (MSE) are as follows:

$$e_{p_1} = 0.004(m), e_{p_2} = 0.003(m), e_{p_3} = 0.007(rad) \quad (4.29)$$

As can be seen, the proposed approach succeeds in controlling very accurately the position and orientation of a point on a deformable rod, despite the different properties of the object. Regardless of the kind of simulated model we use to test the approach, the results show that the proposed method performs very well in simulation. However, the detection technique we employ is the reason why the algorithm performs a little bit better in simulations than in real experiments (in our detecting process, there are certain noises).

4.7 Conclusion

In this chapter, we have proposed a novel approach to address the problem of controlling an LDO. An adaptive control method was developed to move an arbitrary point to a chosen objective without knowing prior model parameters. We analyzed the stability of the controlled system via the Lyapunov theory and presented the adaptation law to update the estimations of the system parameters and the states of the controlled point during the control process. The controller was evaluated using KER and ARAP models in simulation. It was shown that the proposed method is not attached to a specific model with predefined parameters and structures. It was also successfully validated using various objects with different characteristics and flexibility in real experiments. Albeit the MSE of the method presented in this chapter (chapter 4) shows that the accuracy of the method presented in chapter 3 is slightly better (see the results presented in sections 3.6.5 and 4.6.4), it should be noted that the adaptive method presented in this chapter can be applied to a wider range of objects because it does not have the limitations of the previous method (proposed in chapter 3), such as having a predefined model with a fixed of nodes that each node should have a fixed length through the deformation process, or knowing the information of all the nodes in advance.

Although the presented methods in this and the previous chapter are able to control the state of a single point, the challenge of controlling the whole shape of the object was not considered, which is an interesting problem in several domains. To solve this drawback, we will attempt to propose a method to control the whole shape of the object in the next chapters.

Chapter 5

Optimal deformation control framework for elastic linear objects

In this chapter, we pursue our research on the shape controlling of DOs. Our objective is to address the main challenge that is not considered in the previous chapters, which is controlling the entire shape of the objects. It is challenging to find suitable methods to deform the LDOs' entire shape desirably since they have a high-dimensional configuration space. Another challenge in soft object deformation is determining the intended target. To the best of our knowledge, no work to date has investigated this topic, and usually, a predefined target or a shape obtained from the performed experiments has been used. To achieve the declared points, we propose a control framework for changing the shape of LDOs with a robotic arm without knowing the object's properties on a 2-D workspace. In particular, we aim to provide a complete methodology that can be used in the future to manipulate agricultural LDOs, such as branches and twigs of vegetables, without damaging them. The first component of our framework is a shape prediction optimal method that obtains a target shape that minimizes the stress along the target's length. Using this method, the reachability of the target shape can be guaranteed. The second component of our framework is executed later and is based on an indirect optimal controller that automatically drives the objects' shapes into the target shapes by minimizing a cost function that reduces the error between the targets and the current shapes. Several numerical simulations and real experiments are presented to highlight the performance of the proposed methodology.

5.1 Our contributions

The objective of this chapter is to propose an optimal deformation control framework for LDOs. We aim to propose a method to set desired target shapes of LDOs and deform the entire shape of objects toward the proposed desired shapes using an indirect control. The contributions of this chapter are summarized as follows. The first component of our framework is to address the problem of finding desired target shapes for the LDOs. As far as we know, there has been no work on the target shape optimization, and all the previous works had adjusted the target shape based on some former tests [ZNP⁺19, ZNAPC21, NAL13]. However, in this study, we design a method to obtain a reachable target shape that is at minimum stress. This helps to have a stable shape for the target that is less likely to break. It should be mentioned that [BM14] used similarly a minimum-energy-based scheme to model linear objects according to a boundary value problem; however, in this work, we use the optimal theory to find the optimized target shape, and we do not model the object in this way. The second component of our framework is an optimal control strategy to deform LDOs from initial shapes to the obtained target shapes. The method we use is to move mesh nodes on the object's body using the manipulated point

to achieve the desired shape of the object. The proposed method does not need prior knowledge of any dynamical parameters of the object, since we only need to use and trace the surface of the object to define the mesh. Our method is simple to implement, and it does not necessitate collecting the data from a time window to calculate a Jacobian. Moreover, in this chapter, we intend to use a method to calculate a Jacobian that does not have the limitations of the method presented in chapter 3, such as having fixed-length meshes during the entire algorithm execution. The proposed method can control the entire shape of the object in comparison to [ZNF⁺18, NAL17]. We present a wide range of validations, including different simulation tests and robotic experiments with different objects: a thin sponge rod, a plastic rod, and a thick foam rod¹.

5.2 Problem statement

In this chapter, we seek to provide a suitable way to control the entire shape of LDOs. A significant challenge in manipulating these objects is setting a reachable and stable target that is less likely to cause any breakage or failure. Another challenge is to deform them to the targets desirably. These challenges inspire us to propose the current control framework. To clarify the issue, consider the manipulation of the branches in tasks such as tree inspection or harvesting. In these kinds of tasks, the main point is that the exact shape of the target is not specified in advance, and only an estimate of the final pose of the grasped point is known beforehand. In this process, it is essential to move the branches to a secure position that does not break them.

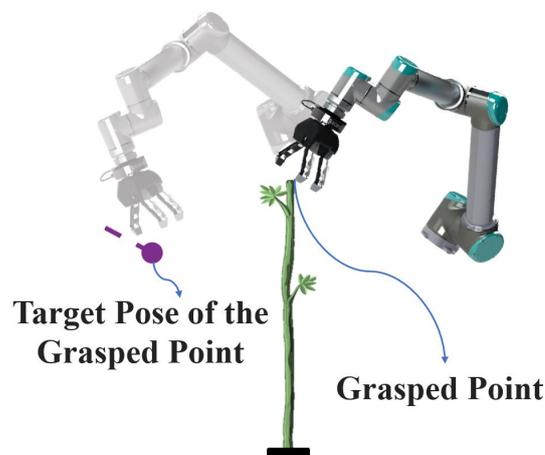


FIGURE 5.1: Desired shape prediction based on the necessitated motion.

The schematic of the studied object with length L is shown in Fig. 5.1. The problem of DO manipulation we deal with in this study can be summarized as follows. Firstly, we seek a method to set a stable and reachable target. To the best of the authors' knowledge, previous studies on the manipulation of DOs have set targets using predefined shapes or pre-performed tests. We develop a method for target shape optimization. Our objective is to minimize stress along the object's length to find the optimized target shape. In this part, there is no need to have exact model values since we aim to deal only with a shape servoing problem. In the next step, the objective is to deform an object toward the obtained target shape without being aware of the object's properties. Therefore, a new optimal controller is adopted to minimize the shape error between the current shape and

¹A video of our simulations and experimental results can be found at: <https://www.youtube.com/watch?v=a3XoJFYlyc4>

the target. Fig. 5.2 shows the overall scheme of the proposed control framework. The current shape of the LDO is shown in Fig. 5.3.

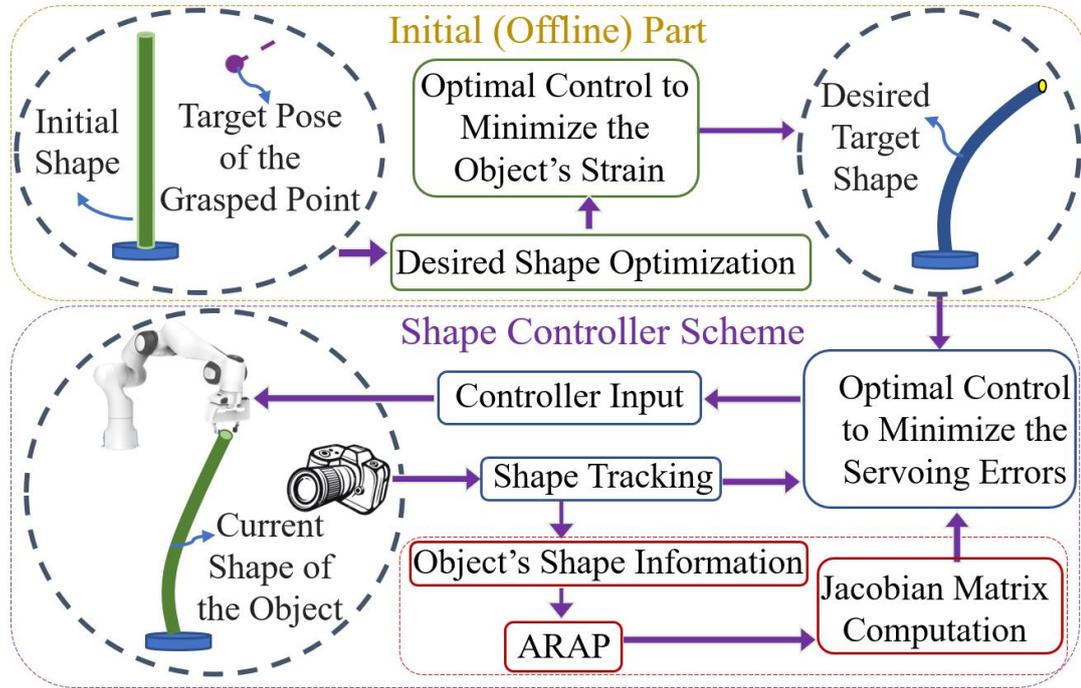


FIGURE 5.2: Schematic diagram of the proposed control framework.

5.3 Desired shape optimization

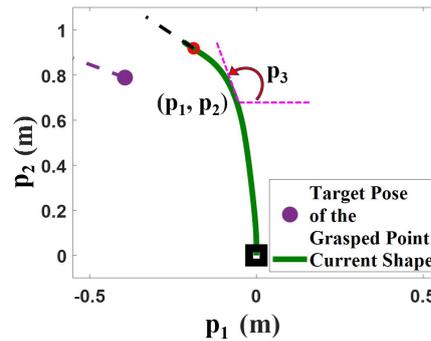


FIGURE 5.3: The shape of the studied object (shown as a green rod), and the final pose of the grasped point (shown as a violet circle with a small dashed line representing its angle). The grasped point is demonstrated as a red circle with a small black dashed line.

It is clear that having a predefined target shape to start the deformation process is not always possible. Accordingly, in the first part of this chapter, we propose a general method to find the desired target shapes of the LDOs. Our goal is to provide a shape that is stable and less likely to break. To solve the mentioned issue, we use a minimum-energy-based scheme to compute the shape of the LDO's target based on the coordination of its endpoints. The full shape of the target can be computed by (see Fig. 5.3)

$$P(s) = [p_1(s), p_2(s), p_3(s)]^T \quad (5.1)$$

The curve P can be computed as a solution to the following stress minimization problem:

$$\begin{cases} \min \int_0^L v^2 ds, \\ [dp_1/ds, dp_2/ds, dp_3/ds]^T = [L \cos(p_3), L \sin(p_3), v]^T, \\ P(0) = P_0, \quad P(L) = P_g \end{cases} \quad (5.2)$$

v represents the stress along the LDO's length [BM14]. $P_0 \in R^3$ and $P_g \in R^3$ are the states of the object at its fixed point and the grasped point (g), respectively (it should be noted that the same methodology as presented can be used even if the two endpoints are manipulable). According to (5.2), we minimize the total stress over the target's length. The problem of shape computation (5.2) can be seen as an optimal control problem. According to the maximum principle theory [PBG62], there exist a costate² trajectory $[f_1, f_2, f_3]^T$ such that ($i = 1, 2, 3$):

$$\dot{p}_i = \nabla_{f_i} H^T, \quad \dot{f}_i = -\nabla_{p_i} H^T, \quad 0 = \nabla_v H^T \quad (5.3)$$

where H is the Hamiltonian function, calculated as follows:

$$H = f_1 L \cos(p_3) + f_2 L \sin(p_3) + f_3 v - e v^2 \quad (5.4)$$

e is a constant bigger than zero. By solving (5.3) based on (5.4), we have (setting $e = 0.5$ [BM14]):

$$\begin{cases} \dot{p}_1 = L \cos(p_3), & \dot{p}_2 = L \sin(p_3), & \dot{p}_3 = v, \\ \dot{f}_1 = 0, & \dot{f}_2 = 0, & \dot{f}_3 = f_1 L \sin(p_3) - f_2 L \cos(p_3), \\ 0 = f_3 - v \end{cases} \quad (5.5)$$

We can rewrite (5.5) in a compact form as

$$\begin{bmatrix} dp_1/ds \\ dp_2/ds \\ dp_3/ds \\ df_1/ds \\ df_2/ds \\ df_3/ds \end{bmatrix} = \begin{bmatrix} L \cos(p_3) \\ L \sin(p_3) \\ f_3 \\ 0 \\ 0 \\ L f_1 \sin(p_3) - L f_2 \cos(p_3) \end{bmatrix} \quad (5.6)$$

The target shape of the object is obtained and can be simulated by solving numerically (5.6) subject to the boundary conditions ($P(0) = P_0$ and $P(L) = P_g$) for different targets of the grasped point.

5.4 Shape control method

In this section, we design a method for controlling the shapes of LDOs toward the target shapes. Let us first define the shape error S_e as:

$$S_e = M_t - M_{tf} \quad (5.7)$$

$M_t \in R^{2n_s}$ denotes the current position of the object and M_{tf} represents its target value. n_s is the number of points (mesh nodes) along the length of the LDO that we use to represent the object's configuration. The controller objective is to drive S_e to zero.

²The costate is related to the state used in optimal control [Lue97]

To design the controller, we set the controller input ($U \in R^3$) as

$$U = \delta G \quad (5.8)$$

where $\delta G = [\delta p_{1g}, \delta p_{2g}, \delta p_{3g}]^\top$. We use a deformation Jacobian ($J \in R^{2n_s \times 3}$) to predict shape variations that relate the robotic arm's motion to the object's body.

$$\delta M_t = J\delta G \quad (5.9)$$

We assume J is a non-singular matrix. To calculate J for LDOs, we exploit the same method as in [SBAMO22], where the ARAP deformation model is used for the shape servoing of planar objects. We assume that the robotic arm grasps the object firmly. It is also assumed that the object's shape is always in quasi-static equilibrium and deforms smoothly under the infinitesimal motion of the robotic arm with elastic behavior. We define a rigid set ($h_i, i = 1, 2, 3$) that contains the three nearest nodes to the grasped point. This set can be identified from the knowledge of the shape and the robotic arm position with respect to the initial shape. A Cartesian frame W is defined to represent the pose of the robotic arm at the grasped point. A mapping w is also defined to find the pose of W from the position of h_i . Using w , one can find the frame's pose $\zeta_i = w(h_i)$. Accordingly, the positions of the nodes of the rigid set can be obtained as $h_i = w^{-1}(\zeta_i)$. The motion of W consists of 3 DOF. Starting from the current shape M_t , we calculate a new ζ_i by infinitesimally perturbing W in each DOF. Using the new $h_i = w^{-1}(\zeta_i)$ at each perturbation, a new stable shape of the object can be obtained by running the simulation of ARAP. Using this new stable shape, δM_t can be calculated, and therefore, we can obtain J corresponding to the perturbed DOF using (5.9). By repeating this procedure for every three DOF, the full matrix J is computed online, and it can be used to design a controller.

The control objective can be formalized as an optimal control problem by defining the following cost function and minimizing it.

$$E = \left(\sum_{j=1}^{t_f} S_e^T(j) Q_s(j) S_e(j) + U^T(j) Q_u(j) U(j) \right) \quad (5.10)$$

By minimizing E , the shape error S_e is also reduced, which means that the object moves towards its target. $t_f \in N$ is the time horizon of the minimization command, and the convergence time to the minimum value of E can be increased or decreased using it. $Q_s(t) \in R^{2n_s \times 2n_s}$ and $Q_u(t) \in R^{3 \times 3}$ are constant diagonal matrices. j represents the discretized time.

Now using (5.9), the discrete state of the system is:

$$M_t(j+1) = M_t(j) + J(j)U(j) \quad (5.11)$$

(5.11) and (5.10) form an optimal controller problem. In this case, the minimizing command is [AM07]

$$U(j) = K(j)S_e(j) \quad (5.12)$$

where $K(j) \in R^{3 \times 2n_s}$ is given by:

$$K(j) = -[J^\top(j)X(j+1)J(j) + Q_u(j+1)]^{-1}J^\top(j)X(j+1) \quad (5.13)$$

X is given by:

$$\begin{cases} X(j) = (X(j+1) - X(j+1)J(j)[J^\top(j)X(j+1)J(j) \\ \quad + Q_u(j+1)]^{-1}J^\top(j)X(j+1)) + Q_s(j), \\ X(t_f) = Q_s(t_f) \end{cases} \quad (5.14)$$

Using (5.12) to (5.14), one can drive S_e to zero, and therefore, the object deforms to the target shape.

5.5 Simulation results

In this section, simulations are carried out to evaluate the performance of the proposed optimal control strategy. Fifty points are assumed along the length of the simulated rod ($n_s = 50$). The length of the object (L) is 1 (m). In each test, the target pose of the grasped point is changed. Then, the target shapes (shown in blue in Fig. 5.4 to 5.6) are obtained using (5.6) at the start of the control scheme. During the simulation, the current shape of the object is continuously simulated and observed using the ARAP model. At each instant, the Jacobian matrix is updated, and then the state of the grasped point (indicated by the red circle in the figures of this chapter) is updated using (5.12). The new shape of the object is obtained using the simulated deformation model. The rod's shape is assumed to remain stable during the entire control process. The computational time of our controller is 52 FPS. The servoing error for the total shape is introduced as:

$$\text{Servoing error} = \sqrt{(1/n_s^2)S_e S_e^T} \quad (5.15)$$

The initial, target, and final shapes of the object in different tests can be observed in Fig. 5.4 to 5.6. The servoing error and the used optimal cost function (5.10) in each test are plotted in Fig. 5.4 to 5.6. Furthermore, it is evident from these figures that the shape error goes to zero as the cost function decreases, which means that the proposed strategy is able to control the object's shape toward its target.

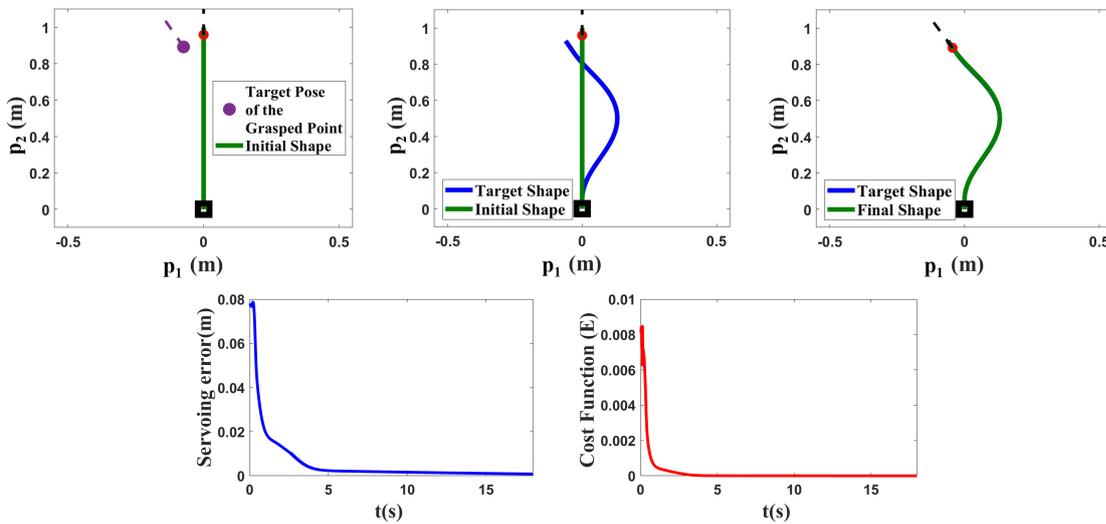


FIGURE 5.4: Simulation 5.1. As shown in the top-left plot, the target pose of the grasped point is set to $[-0.0741, 0.8923, 2.0069]^T$. The object's initial, target, and final shapes can be observed in the first row. The servoing error and the cost functions of the controller (E) are plotted in the second row.

5.6 Robotic experiments

We test our algorithm using a robotic manipulation system at Pascal Institute in Clermont Auvergne University (UCA). The experiments are performed using a UR10 arm with a

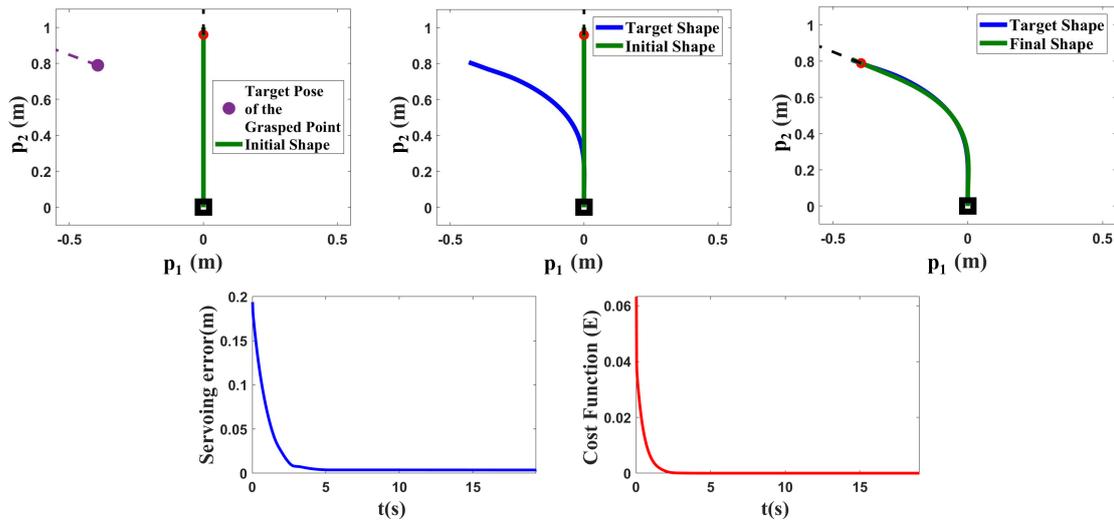


FIGURE 5.5: Simulation 5.2. As shown in the top-left plot, the target pose of the grasped point is set to $[-0.3941, 0.7890, 2.6384]^T$. The object's initial, target, and final shapes can be observed in the first row. The servoing error and the cost functions of the controller (E) are plotted in the second row.

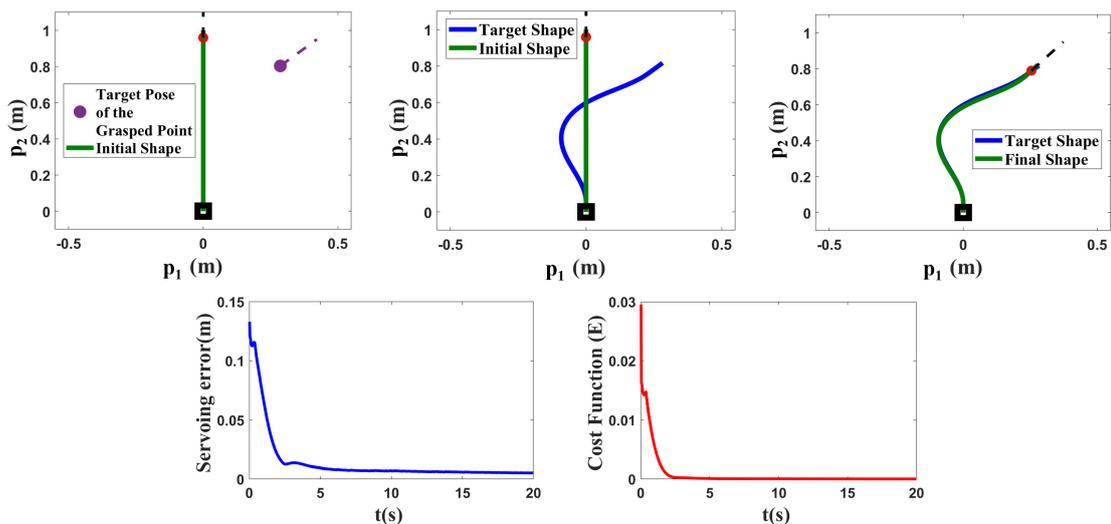


FIGURE 5.6: Simulation 5.3. As shown in the top-left plot, the target pose of the grasped point is set to $[0.2861, 0.8030, 0.8249]^T$. The object's initial, target, and final shapes can be observed in the first row. The servoing error and the cost functions of the controller (E) are plotted in the second row.

three-finger gripper. Our vision system includes a fixed camera that can provide color images and depth at 30 FPS³. An edge detection algorithm has been used to obtain the position of the points along the object's length using the OpenCV library of C++. Due to the used detection method, there exists some noise in our results. In our experiments, we use a uniform discretization to generate the points on the object. Fifty points are used to discretize the object (i.e., $n_s = 50$). ROS melodic connects the controller and detection nodes to the arms. The experimental setup is shown in Fig. 5.7.

We now present three example tasks where the proposed method is effective, and the results are shown in Fig. 5.8 to Fig. 5.10. In our experiments, we first obtain the

³Frames per second

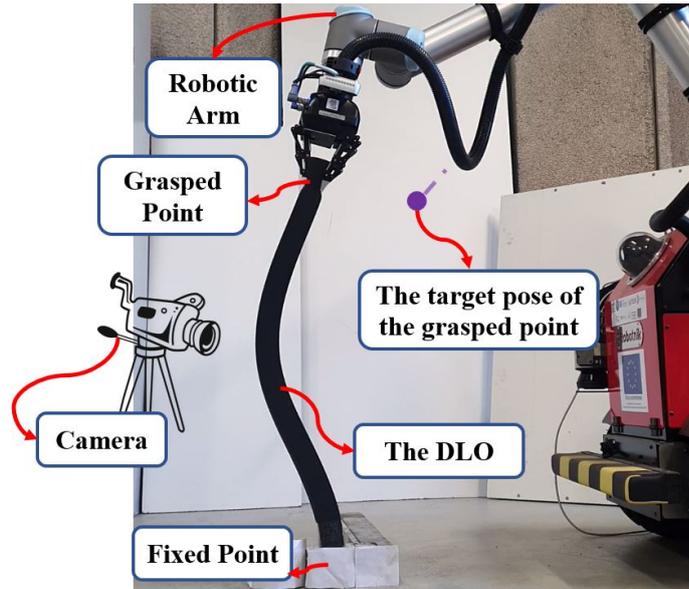


FIGURE 5.7: The experimental setup used in this work.

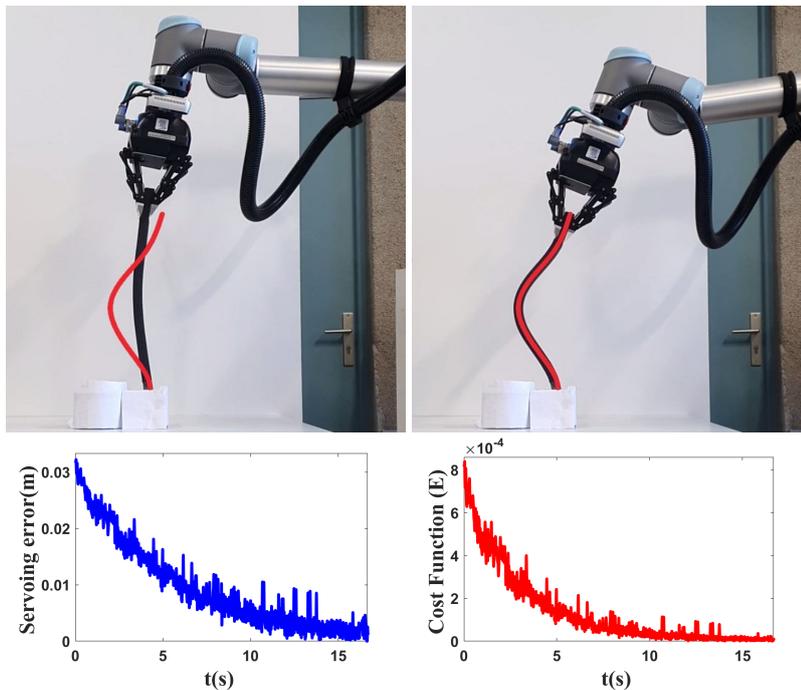


FIGURE 5.8: Experiment 5.1. The target pose of the grasped point is set to $[0.0466, 0.3691, 1.2392]^T$. The object's initial, target, and final shapes can be observed in the first row. The servoing error and the cost functions of the controller (E) are plotted in the second row.

entire shape based on the target position of the grasped point using (5.6), and then we start the deformation process. The Jacobian matrix is obtained at each iteration using the information of the current shape, and then the states of the grasped point are updated using the control law (5.12) at each instant. We use three different LDOs with various flexibility to conduct the experiments: a small deformable sponge rod with the length of $0.43(m)$, a flexible rod made of plastic with the length of $0.89(m)$, a deformable foam rod with the length of $0.87(m)$. The objects' initial, target, and final shapes can be found in the

first row of Fig. 5.8 to Fig. 5.10. The target shapes are shown in red. The servoing error and the used optimal cost function (5.10) in each experiment are plotted in the second row of Fig. 5.8 to Fig. 5.10. It is discernible that the servoing errors have converged to zero by using the proposed algorithm during the deformation, and the presented method achieves its objective.

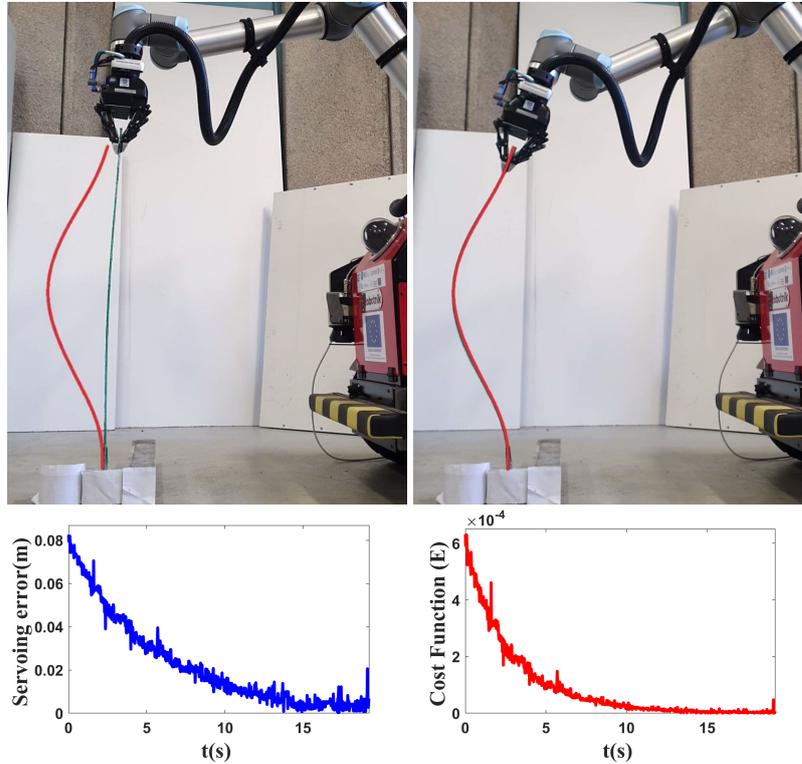


FIGURE 5.9: Experiment 5.2. The target pose of the grasped point is set to $[-0.0383, 0.8395, 1.2654]^T$. The object's initial, target, and final shapes can be seen in the first row. The servoing error and the cost functions of the controller (E) are plotted in the second row.

TABLE 5.1: Final error of the performed tests.

Test number	Final servoing error (m)
Sim. 5.1	0.001
Sim. 5.2	0.003
Sim. 5.3	0.005
Exp. 5.1	0.004
Exp. 5.2	0.009
Exp. 5.3	0.006

5.6.1 Mean squared error (MSE)

We will present the mean squared error (**MSE**) of all tests in this section to show the effectiveness of the proposed technique. To determine the mean errors between the target and final shape in the performed tests, **MSE** is defined as:

$$e_s = \sqrt{\frac{1}{n} \sum_{r=1}^n (\Delta s_r)^2} \quad (5.16)$$

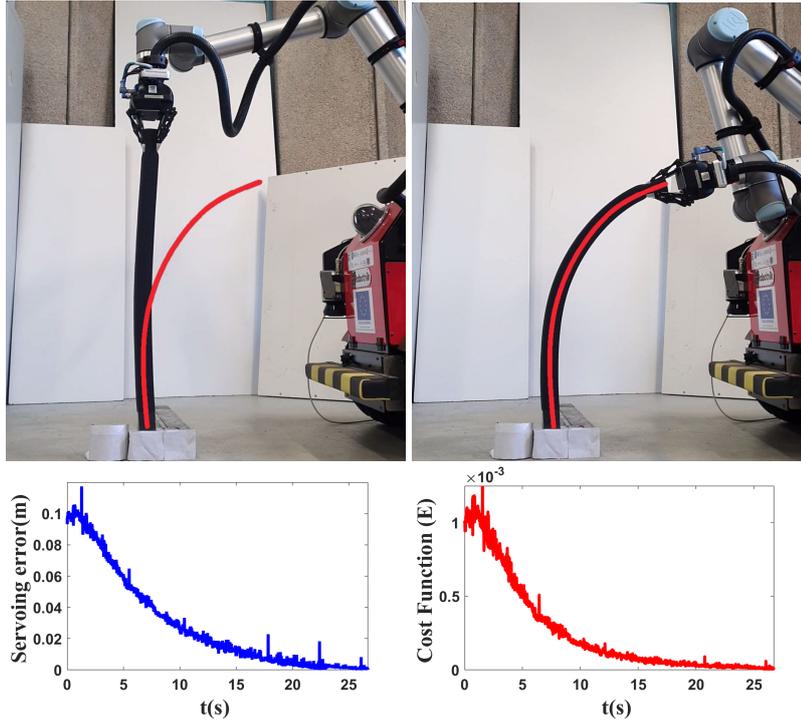


FIGURE 5.10: Experiment 5.3. The target pose of the grasped point is set to $[0.3262, 0.7232, 0.5585]^T$. The object's initial, target, and final shapes can be seen in the first row. The servoing error and the cost functions of the controller (E) are plotted in the second row.

n is the number of tests, Δs_r is the final servoing error of each test. Thus, for the simulation tests ($n = 3$) and based on the results presented in Table 5.1, we have:

$$e_s = 0.003(m) \quad (5.17)$$

The same analysis is performed on the experimental tests and the outcome are is follows:

$$e_s = 0.006(m) \quad (5.18)$$

As shown, the proposed approach is able to precisely control the shape of the adopted objects despite their various properties. The results presented in Table 5.1 show that the presented method works accurately in simulation regardless of the type of simulated model we use to validate the method. However, the algorithm works slightly better in simulations than in experiments due to our detection method (as can be seen in Fig. 5.8 to Fig. 5.10, some small vibrations exist in our experimental results).

5.7 Conclusion

In this chapter, we presented a control framework to control LDO's shapes. The first module of our framework provides a target shape by minimizing the stress along the object's length. The reachability of the target shape is checked in simulation using the proposed algorithm. The second module of the methodology is an optimal controller that is presented to reshape the LDOs toward the target shapes. An online Jacobian calculation is used to calculate the relation between the shape of the LDOs and the motion of the robotic arm. It should be noted that compared to the method presented in chapter

3, the Jacobian calculation of this chapter is more complicated, and the computational time is a little bigger (see Appendix A), since we need to compute at each time iteration. However, this method does not have the limitations of the method presented in chapter 3, such as having fixed-length meshes without sudden deformations. Furthermore, this method can be extended to other types of objects (such as 2-D or 3-D shapes) with the same formula. The proposed method works in real-time, and it controls the entire shape of the object. The controller was evaluated using different simulations and experiments on different objects. However, if the Jacobian calculation cannot be obtained, this method cannot be used for the studied objects. Another limitation of this method is that it controls the 2-D shape of the object. Thus, in the next chapter, we will continue to study the manipulation of DOs while trying to propose a method that does not require a deformation Jacobian based on a predefined model to control the shape in 3-D. This leads us to suggest a way that can be applicable for a wide range of objects.

Chapter 6

3-D shape control of LDOs using an adaptive Lyapunov-based scheme

In this chapter, we aim to take a step further than the previous chapters and focus on the control of the entire shape of LDOs in 3-D, with agriculture as the primary target. As previously mentioned, there are several challenges in manipulating LDOs in agriculture, i.e., the variety of objects in this field is wide, and the deformation properties of objects cannot be easily recognized in advance. Thus, in this chapter, we address the manipulation of the entire shape of LDOs by considering these challenges. Concretely, we extend the adaptive control method proposed in chapter 4 to manipulate LDOs by controlling their shape in 3-D space towards previously defined targets. Our method can follow a desired dynamic evolution of the shape with a smooth deformation that brings about a stable gripper motion. Using this method, obtaining any form of offline Jacobian or having a model is not necessary, which makes it more feasible to control the shape of a broader range of objects. To investigate this fact, we demonstrate the effectiveness of our proposed control scheme through different robotic experiments on various plant branches and a foam rod.

6.1 Introduction

The main goal of the current work is to extend the previous shape servoing approaches to agriculture applications [Dav06, YSF⁺20, BPG⁺17]. The main challenges are: first, the diversity of objects in this field is vast and consequently a specific model cannot be used; second, determining the properties of objects is not always possible; and third, objects are mostly very fragile and a stable robotic motion is essential. The state-of-the-art methods previously described in chapter 2 had not considered the aforementioned challenges in this field [LKM20, QMZ⁺21, KFB⁺21, NAYW⁺16].

One of the very first works in agriculture that assumed robots are dealing with DOs can be found in [TKK01]. In this study, the object is assumed to be a rod modeled using FEM. However, this work mainly focused on obtaining equations of a multiple mobile manipulator system, and no shape controllers were presented. Inspired by that work, we assume that the branches can be considered as LDOs, as shown in Fig. 6.1. However, we do not need an FEM representation of the LDOs as in [TKK01] but only a set of controllable points. In the previous chapters, we attempted to control an arbitrary point along the length of the stems and branches and control LDOs' shape in 2-D. In this study, we intend to extend the work presented in chapter 4 to control the entire shape of the object in 3-D space.

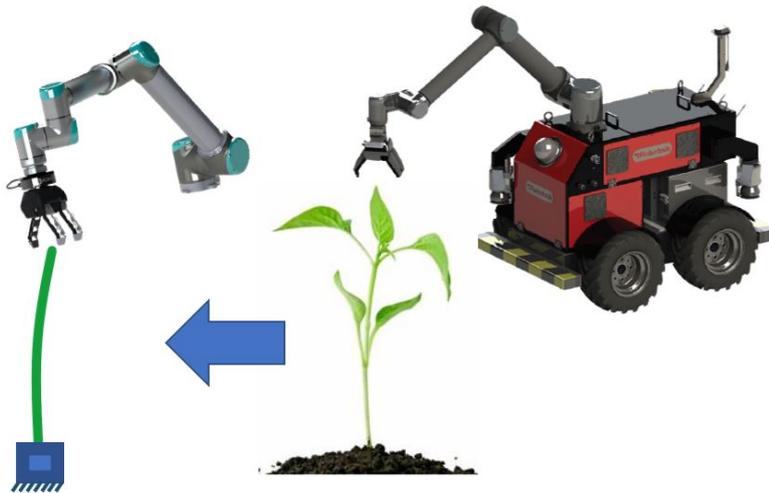


FIGURE 6.1: Approximating the DO, inspired by [TKK01].

6.1.1 Our contribution

This chapter proposes an indirect model-free deformation method of LDOs in 3-D space to be used in the agricultural field. Apart from agriculture, the proposed method can also be used in other domains. We aim to control the full shape of an LDO using merely a robotic gripper grasping the object at one of its ends (i.e. indirect control). This is done without knowing the object's properties (i.e. model-free). An adaptive control strategy is proposed to achieve the stated objective. Our contributions can be listed as follows:

- We extend the adaptive method introduced in chapter 4 to deform the shape of LDOs in 3-D space. In particular, the proposed method is suitable for manipulating stems or branches for pruning, plant inspection, or fruit harvesting operations.
- Our method works in real-time and is suitable for large deformations. Our proposed method controls the full shape of the object and not a simplified representation of it.
- In contrast to [LWL19] and related adaptive deformation control approaches [ZNF⁺18, LKM20, NAL17], the proposed method can be used to track a desired dynamic evolution of the entire shape of the object, rather than simply drive fixed set-points to their corresponding targets. This gives a better control over the evolution path of the object and the completion time of the task. It also provides a smooth deformation since we track time-varying trajectories instead of fixed distant targets, which helps to have a deformation without sudden changes. This can protect the object from possible damages.
- The proposed method does not require any prior information regarding object's deformation. In addition, we do not need to compute any offline Jacobian matrix to know how displacements of the manipulator are mapped to deformations of the object.

6.2 Problem definition

As previously described, there are many objects in agriculture that can deform like an elastic linear rod (such as branches and twigs of plants and trees). Therefore, we consider these objects as LDOs, and we put our effort into designing a method for controlling their shapes. We adopt p_i ($i \in \{1, \dots, n\}$) (i.e., the six coordinates -position and orientation- of every point on the LDO surface) to show the complete state of the object. Since we deal

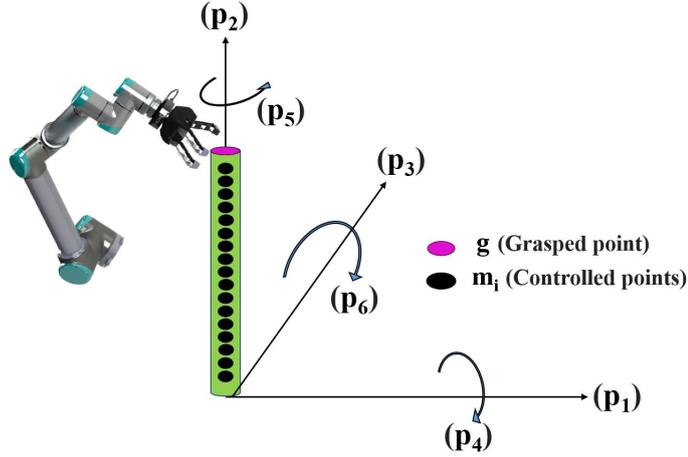


FIGURE 6.2: Representation of the considered LDO in the current study.

with plants, we presume one of the ends is fixed (connected to the ground by the roots), while the other end is grasped by the robot, as illustrated in Fig. 6.2.

We represent the LDO discretely with a set of $n + 3$ nodes. The nodes are ordered consecutively starting from the fixed end of the object, and they sample the object's length (L) uniformly. It is assumed that we can obtain the position of each node at its center during the manipulation. A sensor (an Intel RealSense 3-D camera) is used in our experiments for this purpose. We define the controlled points along the length of the object in such a way that the position of each point is the average position of the center of its two closest nodes. We use each controlled point's closest nodes to obtain its full state $[p_{1m_l}, p_{2m_l}, p_{3m_l}, p_{4m_l}, p_{5m_l}, p_{6m_l}]^T$ (position and orientation in 3-D). Therefore, we have $n + 2$ controlled points, indexed $0, \dots, n + 1$. We dedicate controlled points 0 and $n + 1$ to the fixed and gripped parts, respectively (this means that these two points are no longer controlled points). Without loss of generality, we assume that using a robotic gripper, we are able to deform the whole object by controlling the state of the endpoint. G is the state of the object at the grasped point g (where $G = [p_{1g}, p_{2g}, p_{3g}, p_{4g}, p_{5g}, p_{6g}]^T$). p_{im_l} represent the states of any point along the length of the object that is not being gripped (i.e., the controlled (servoed) points m_l). We use l to show the servoed point's number, where $l \in \{1, \dots, n\}$. We define M_l as the state of m_l , which contains the information of the point number l along the object's length.

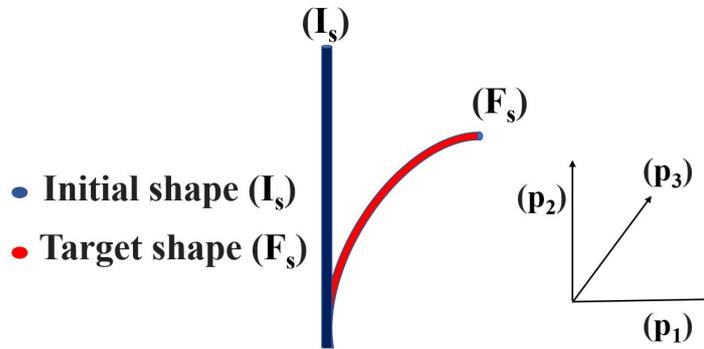


FIGURE 6.3: The initial (I_s) and target (F_s) shape of the studied object.

Our objective is to deform the object from an initial shape I_s to a target shape F_s (shown in Fig. 6.3). To this end, we should drive the error Er_l for every l to zero where

Er_l is defined as:

$$Er_l = M_l - M_{lf} \quad (6.1)$$

where the set of M_{lf} expresses the target shape of the controlled points m_l . Er_l indicates the error between the current and the target shapes. By leading this error to zero, the object is deformed to the target shape.

TABLE 6.1: Variables used in the proposed controller scheme

$M_l = [p_{1m_l}, p_{2m_l}, p_{3m_l}, p_{4m_l}, p_{5m_l}, p_{6m_l}]^T$	State of the controlled points $l \in \{1, \dots, n\}$
$M_s = [p_{1s}, p_{2s}, p_{3s}, p_{4s}, p_{5s}, p_{6s}]^T$	Average of the controlled points states
$M_{sf} = [p_{1sf}, p_{2sf}, p_{3sf}, p_{4sf}, p_{5sf}, p_{6sf}]^T$	Average of the target points states
$M_{sdes} = [p_{1sdes}, p_{2sdes}, p_{3sdes}, p_{4sdes}, p_{5sdes}, p_{6sdes}]^T$	Desired trajectories
$\tilde{M}_s = [\tilde{p}_{1s}, \tilde{p}_{2s}, \tilde{p}_{3s}, \tilde{p}_{4s}, \tilde{p}_{5s}, \tilde{p}_{6s}]^T$	Tracking errors
P_i	Vectors of the states $i \in \{1, \dots, 6\}$
\hat{C}_i	Adaption laws
\hat{C}_i	Vectors of the estimation of the parameters

6.3 Control scheme

In this section, we develop a controller to drive every Er_l to zero. Since the gripper fixes the velocity of the grasped point, the controller input $U = [u_1, u_2, u_3, u_4, u_5, u_6]^T$ is set as follows:

$$U = \dot{G} \quad (6.2)$$

For the set of points that we use to represent the object's shape, there is a relation between their displacements and also their velocities with the motion of the gripper \dot{G} [NALRL14, Ogd97].

$$M_l = f_1(\dot{G}) \quad \& \quad \dot{M}_l = f_2(\dot{G}) \quad (6.3)$$

Accordingly, one can write:

$$\dot{G} = F(M_l, \dot{M}_l) \quad (6.4)$$

To provide a method to move these points to their corresponding targets, we assume a relation (i.e., (6.4)), between the current state of the points along the length of the object (p_{im_l}) and the controller inputs (u_i , where $i \in \{1, \dots, 6\}$) at each time instant.

This study deals with an underactuated system since the number of actuated **DOF** is lower than the number of controlled **DOF**. To solve this problem, instead of controlling each point, we attempt to control a new variable $M_s = [p_{1s}, p_{2s}, p_{3s}, p_{4s}, p_{5s}, p_{6s}]^T$, where p_{is} (as shown below in (6.5)) represents the average of the state (position and orientation) of all the controlled points p_{im_l} :

$$p_{is} = \frac{\sum_{l=1}^n p_{im_l}}{n} \quad (6.5)$$

In other words, M_s is the variable we use to represent the full state of the object. By controlling it (i.e., M_s), we are able to control the full shape. This is analogous to standard control strategies in underactuated control systems, and in particular, to some existing

works in DO shape control [NALRL13a, ZNF⁺18]. In these works, the system is only locally stable with respect to a fixed desired state. For defining the angles, we assume that the object will not be faced with sudden and significant changes of orientation in each time step.

Therefore, in this section, an adaptive control method is designed to drive M_s toward its target, which results in driving each Er_l for all servoed nodes $l \in \{1, \dots, n\}$ to zero. To do that, we draw inspiration from model-free schemes that have been proposed for DO manipulation [NAL17, ZNF⁺18]. However, to design our method, we also use some additional terms. These terms allow us to track a desired trajectory of the state, instead of just driving a group of points to their corresponding fixed targets. The proposed controller formulation follows a well-known adaptive control scheme presented in [SL⁺91, ASTZ17, ASTZ18]. Therefore, in this study, for each possible desired shape of the object, the following relation between the grasped point g and p_{is} is assumed:

$$\dot{p}_{ig} = a_{i0}\dot{p}_{is} + \Omega_i\tilde{M}_s \quad (6.6)$$

where Ω_i is a 1×6 vector containing some values related to the controlled points (namely a_{ij} ($j \in \{1, \dots, 6\}$)). We define a_{i0} as a non-zero positive parameter which permits to account for the speed of p_{is} during the process. \tilde{M}_s represents the tracking errors and is introduced as:

$$\tilde{M}_s = M_s - M_{sdes} \quad (6.7)$$

In (6.7), $M_{sdes}(t)$ is the desired trajectory of M_s at each time step (t) of the deformation process. p_{isdes} is the i th component of the M_{sdes} , as presented in Table 6.1. We intend to develop an adaptive algorithm to track these trajectories, defined as bounded and differentiable. To achieve a smooth and uniform dynamic evolution of the LDO's shape, the following trajectories are proposed to be used as the desired position trajectories for each point:

$$p_{imides} = (p_{im_i0} - p_{im_if}) \exp(-\kappa t) + p_{im_if} \quad (6.8)$$

p_{im_if} is the target and p_{im_i0} is the initial value of the p_{im_i} . The constant parameter κ (the exponential coefficient) stands for a predefined positive constant. The used exponential trajectories bring about a good control over the completion time of the task. They link implicitly all the points m_i by making them evolve in the same exponential fashion. In our earlier works [ASF20], these types of trajectories (i.e., exponential function) were examined in similar control studies, and they functioned effectively and were applicable. From (6.8), we obtain p_{isdes} based on the average of all the desired points p_{imides} and their angles.

For each $i \in \{1, \dots, 6\}$, (6.6) can be rewritten to obtain the following equation:

$$\dot{p}_{ig} = a_{i0}\dot{p}_{is} + \sum_{j=1}^6 a_{ij}\tilde{p}_{js} = R_i C_i^T \quad (6.9)$$

where R_i are vectors that have the following form:

$$R_i = [\dot{p}_{is}, \tilde{p}_{1s}, \tilde{p}_{2s}, \tilde{p}_{3s}, \tilde{p}_{4s}, \tilde{p}_{5s}, \tilde{p}_{6s}] \quad (6.10)$$

where \tilde{p}_{is} is $p_{is} - p_{isdes}$. The vectors C_i contain unknown parameters:

$$C_i = [a_{i0}, a_{i1}, a_{i2}, a_{i3}, a_{i4}, a_{i5}, a_{i6}] \quad (6.11)$$

where a_{ik} ($k \in \{0, 1, \dots, 6\}$) are parameters that depend on the current and desired shape of the object, which are representative of the rod deformation. We presume that these parameters are approximately constant in a local neighborhood of the current shape at each time instant. They must be estimated online to find the relation between the motion of the gripper and M_s .

By defining the controller based on the expression (6.9) and using p_{isdes} , we can track the deformation evolution closely while converging to the target shape. This characteristic of the proposed method can be helpful to have a smooth deformation without sudden movement, which is essential to protect the object against possible damages. This trajectory also allows reaching the desired target of M_s in a limited time. Furthermore, by using the proposed method of trajectory tracking of a time-varying desired state, we are not limited to operating in a local neighborhood of a fixed desired state (as in [ZNF⁺18, NALRL13a]). Indeed, our approach is capable of controlling larger, non-local deformations, associated with the defined time evolution of the desired state since at each time instant, we can assume that the system is in a local neighborhood of the desired value.

6.3.1 Adaptive controller design

This section aims to develop an adaptive Lyapunov-based technique to find estimations a_{ik} of the parameters of (6.11), and define a control law based on these estimated parameters. The block diagram of the proposed controller is shown in Fig. 6.4.

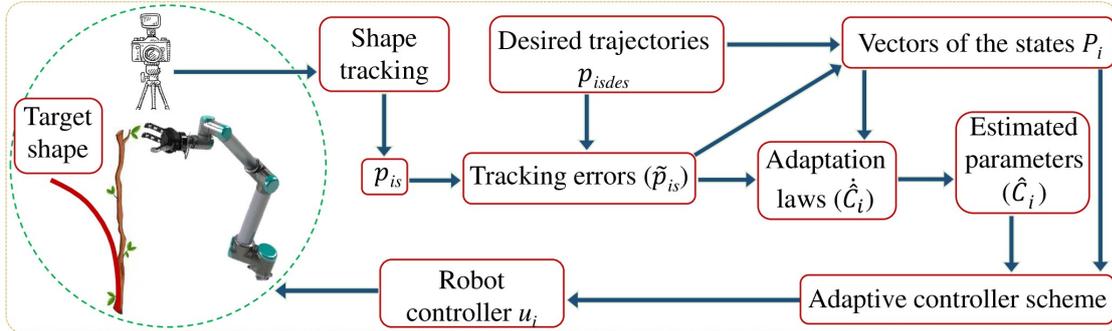


FIGURE 6.4: Schematic diagram of the proposed control strategy.

The goal is to design a controller that can deform the object based on desired trajectories. To this end, we change the form of the speed of \dot{p}_{is} in (6.9) to $\dot{p}_{isdes} - \alpha \tilde{p}_{is}$, where α is a positive constant. This helps to generate desired trajectories for the velocities to be used in the controller design. To find the estimation of a_{ik} , we replace their estimations (\hat{a}_{ik}) in (6.6) which will be updated using an adaptation law. We use the accent $\hat{\cdot}$ for the estimated parameters of the system. Therefore, the adaptive control law u_i is defined as:

$$u_i = \hat{a}_{i0}(\dot{p}_{isdes} - \alpha \tilde{p}_{is}) + \sum_{j=1}^6 \hat{a}_{ij} \tilde{p}_{js} \quad (6.12)$$

As explained in the introduction, most of the works in this area have been done by estimating the Jacobian matrix or having an object model, while using the adaptive controller (6.12) with the proposed scheme, we do not need the mentioned information.

To rewrite (6.12) in a similar form as (6.9), we replace R_i and C_i with new forms P_i and \hat{C}_i . Therefore, (6.12) can be rewritten as:

$$u_i = P_i \hat{C}_i^T \quad (6.13)$$

where:

$$P_i = [(\dot{p}_{isdes} - \alpha \tilde{p}_{is}), \tilde{p}_{1s}, \tilde{p}_{2s}, \tilde{p}_{3s}, \tilde{p}_{4s}, \tilde{p}_{5s}, \tilde{p}_{6s}] \quad (6.14)$$

\hat{C}_i are the vectors of system parameters' estimations:

$$\hat{C}_i = [\hat{a}_{i0}, \hat{a}_{i1}, \hat{a}_{i2}, \hat{a}_{i3}, \hat{a}_{i4}, \hat{a}_{i5}, \hat{a}_{i6}] \quad (6.15)$$

The adaption laws for updating the estimated parameters can be expressed as:

$$\dot{\hat{C}}_i = -P_i T_i \tilde{p}_{is} \quad (6.16)$$

where the adaptation gains T_i are constant non-singular positive definite matrices of size 7×7 . \tilde{p}_{is} , introduced earlier, is the i th element of \tilde{M}_s , as presented in Table 6.1. The adaption laws will be analyzed in the next section using the Lyapunov formalism.

6.3.2 Lyapunov stability investigation

In this section, we use the Lyapunov theorem [SL+91] to investigate the validity of the proposed adaptive control (6.12) and adaptation laws (6.16). To this end, the closed-loop dynamics of the system are obtained. Doing so, we substitute the control law (6.12) in the relation between the object's shape and grasped point (6.9). By performing several simple mathematical operations, we have:

$$\begin{aligned} & a_{i0} \dot{p}_{is} + \sum_{j=1}^6 a_{ij} \tilde{p}_{js} = \\ & \hat{a}_{i0} (\dot{p}_{isdes} - \alpha \tilde{p}_{is}) + \sum_{j=1}^6 \hat{a}_{ij} \tilde{p}_{js} + a_{i0} (\dot{p}_{isdes} - \alpha \tilde{p}_{is}) - a_{i0} (\dot{p}_{isdes} - \alpha \tilde{p}_{is}) \end{aligned} \quad (6.17)$$

By reordering the terms in (6.17), one can write:

$$a_{i0} (\dot{p}_{is} - (\dot{p}_{isdes} - \alpha \tilde{p}_{is})) = (\dot{p}_{isdes} - \alpha \tilde{p}_{is}) (\hat{a}_{i0} - a_{i0}) + \sum_{j=1}^6 \tilde{p}_{js} (\hat{a}_{ij} - a_{ij}) \quad (6.18)$$

Using equations (6.11), (6.14), and (6.15) in (6.18), the system's closed-loop dynamics for every i can be found as:

$$a_{i0} (\dot{\tilde{p}}_{is} + \alpha \tilde{p}_{is}) = P_i \tilde{C}_i^T \quad (6.19)$$

where $\tilde{C}_i = \hat{C}_i - C_i$. By simplifying (6.19), the closed-loop dynamics of the controllers for every i are reformulated as:

$$\dot{\tilde{p}}_{is} = -\alpha \tilde{p}_{is} + \frac{1}{a_{i0}} P_i \tilde{C}_i^T \quad (6.20)$$

A positive definite Lyapunov candidate is used as follows to analyze the system stability and the tracking convergence using the proposed controller.

$$V = \frac{1}{2} \left(\sum_{i=1}^6 \tilde{p}_{is}^2 + \frac{1}{a_{i0}} \tilde{C}_i T_i^{-1} \tilde{C}_i^T \right) \quad (6.21)$$

The time derivative of V is determined as:

$$\dot{V} = \sum_{i=1}^6 (\dot{\tilde{p}}_{is} \dot{\tilde{p}}_{is} + \frac{1}{a_{i0}} \dot{\tilde{C}}_i T_i^{-1} \tilde{C}_i^T) \quad (6.22)$$

where as explained earlier, we neglect $\dot{\tilde{C}}_i$ compared to $\dot{\tilde{C}}_i$, so $\dot{\tilde{C}}_i = \dot{\tilde{C}}_i$. Therefore, via (6.20), (6.22) is transformed into:

$$\dot{V} = \sum_{i=1}^6 (-\alpha \tilde{p}_{is}^2 + \frac{1}{a_{i0}} P_i \tilde{C}_i^T \tilde{p}_{is} + \frac{1}{a_{i0}} \dot{\tilde{C}}_i T_i^{-1} \tilde{C}_i^T) \quad (6.23)$$

Using the parameters' adaptation laws (6.16), the time derivative of the Lyapunov function (6.23) finally has the following expression:

$$\dot{V} = \sum_{i=1}^6 -\alpha \tilde{p}_{is}^2 \leq 0 \quad (6.24)$$



FIGURE 6.5: Different used objects in the performed experiments: a foam rod ($L = 0.87$ (m)), three branches of different plants ($L = 0.40, 0.60, 0.38$ (m))

Since α is a positive constant, the time derivative of the Lyapunov function is negative semi-definite. It should be noted that the only way to have $\dot{V} = 0$ is to have all errors, i.e., \tilde{p}_{is} for all $i \in \{1, \dots, 6\}$, equal to zero. According to (6.21), the Lyapunov function V is positive definite in terms of \tilde{p}_{is} and \tilde{C}_i . The time derivative of Lyapunov function in (6.23) is negative semi-definite $\dot{V} \leq 0$. Therefore, V is bounded and consequently \tilde{p}_{is} and \tilde{C}_i remain bounded. Based on Barbalat's lemma [SL⁺91], if w is a uniformly continuous function and the limit of the integral $\lim_{t \rightarrow \infty} \int_0^t w(\eta) d\eta$ exists and has a finite value, it is concluded that:

$$\lim_{t \rightarrow \infty} w(t) = 0 \quad (6.25)$$

Now, considering $w(t) = \sum_{i=1}^6 \alpha \tilde{p}_{is}^2$, \dot{V} in (6.24) can be written as

$$\dot{V} = - \sum_{i=1}^6 \alpha \tilde{p}_{is}^2 = -w(t) \quad (6.26)$$

By integrating both sides of (6.26) from $t = 0$ to $t \rightarrow \infty$, it is obtained:

$$V(0) - V(\infty) = \lim_{t \rightarrow \infty} \int_0^t w(\eta) d\eta \quad (6.27)$$

As shown in (6.24), $\dot{V} = dV/dt \leq 0$. Therefore, $V(0) - V(\infty)$ is positive and finite. Accordingly, $\lim_{t \rightarrow \infty} \int_0^t w(\eta) d\eta$ in (6.27) exists and has a finite positive value because of the positiveness of $w(t)$. As a result, based on Barbalat's lemma [SL⁺91], one can write:

$$\lim_{t \rightarrow \infty} w(t) = \lim_{t \rightarrow \infty} \sum_{i=1}^6 \alpha \tilde{p}_{is}^2 = 0 \quad (6.28)$$

Since $\tilde{p}_{is}^2 \geq 0$ and α is a positive non-zero constant, (6.28) ensures the convergence to $\tilde{p}_{is} \rightarrow 0$ as $t \rightarrow \infty$ for all i . Thus, the proposed adaptive controller achieves its objective which is tracking the desired trajectories ($p_{is} \rightarrow p_{isdes}$). The variable M_s containing all the p_{is} tracks the desired trajectory, which means the tracking error \tilde{M}_s (6.7) goes to zero. Consequently, the shape error decreases and reaches zero.

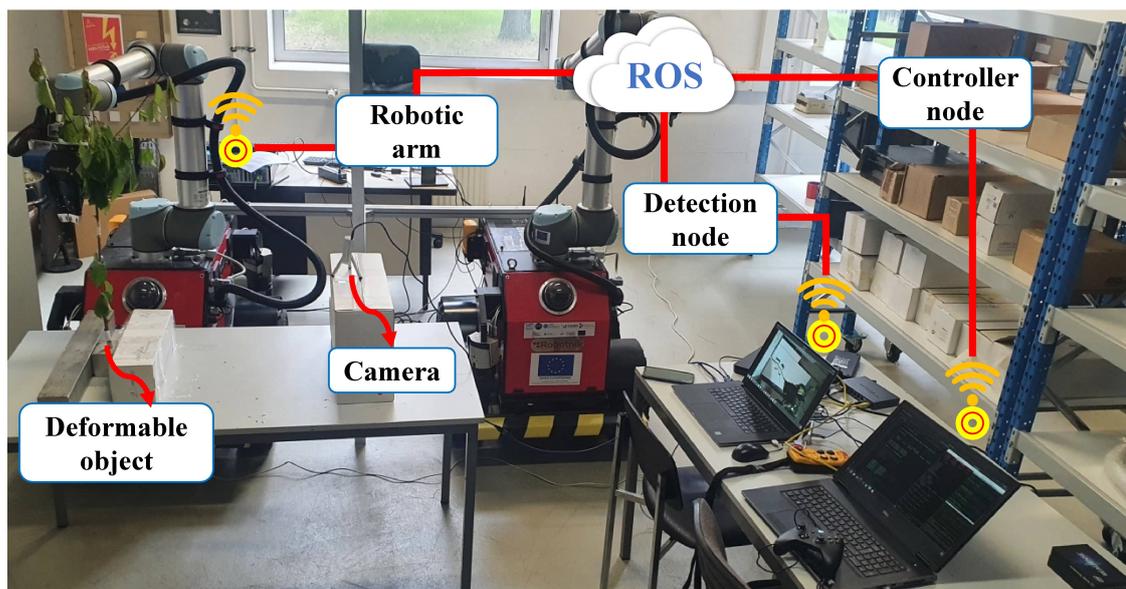


FIGURE 6.6: The experimental setup used in this work.

6.4 Experimental results

We validate the performance of the proposed adaptive control strategy through various experiments on real DOs. The objects of interest are a foam rod and three different plants, as shown in Fig. 6.5. We choose objects with different length and stiffness, to validate the capability of our control strategy to handle objects with diverse and unknown characteristics. The experiments are performed with a UR10 arm of a mobile manipulator¹. We use a novel tracking pipeline for tracking the object deformation in real-time. This tracking pipeline works based on a 3-D template of the object created offline. The template is a 3-D mesh of the object at its rest shape; a rectangular cuboid for the foam rod and a cylinder for the plants. The resolution of the meshes along the object is chosen in such a way that we have 20 controlled points (i.e., $n = 20$). We use an Intel RealSense 3-D camera to provide a point cloud of the scene as the input for the tracking pipeline. The camera is externally calibrated with the robot. The tracking process starts by setting the object in a known shape in front of the camera and triggering the tracking. The tracking pipeline uses the inferred shape of the object in the previous frame and the point cloud captured in the current frame to infer the object shape in the current frame. This is done by: (i) using an intensity filter to filter the pixels and, consequently, the point cloud belonging to the object, (ii) applying an ICP algorithm to rigidly transfer the object mesh

¹A video of our experimental results can be found at: https://www.youtube.com/watch?v=FIEMAy_IcZo

to the point cloud, (iii) applying an ICP-like algorithm proposed in [PLS15] to find correspondences between the object and the point cloud, and (iv) applying a deformation constraint using position-based dynamics [MHHR07a] to the mesh of the object. The last step deforms the object according to the correspondences found in the step (iii). With the tracking pipeline running, we deform the object to a desired shape. We save this desired shape to be used later by the control method. We then deform the object to its initial shape. We, finally, start the controller to drive the object from the initial shape to the desired shape. The tracking pipeline and the controller are both written in C++ and run on two standard Dell laptops, each with an Intel Core i7 CPU. We use ROS Melodic to connect our controller and tracking nodes to the arm. The experimental setup is shown in Fig. 6.6.

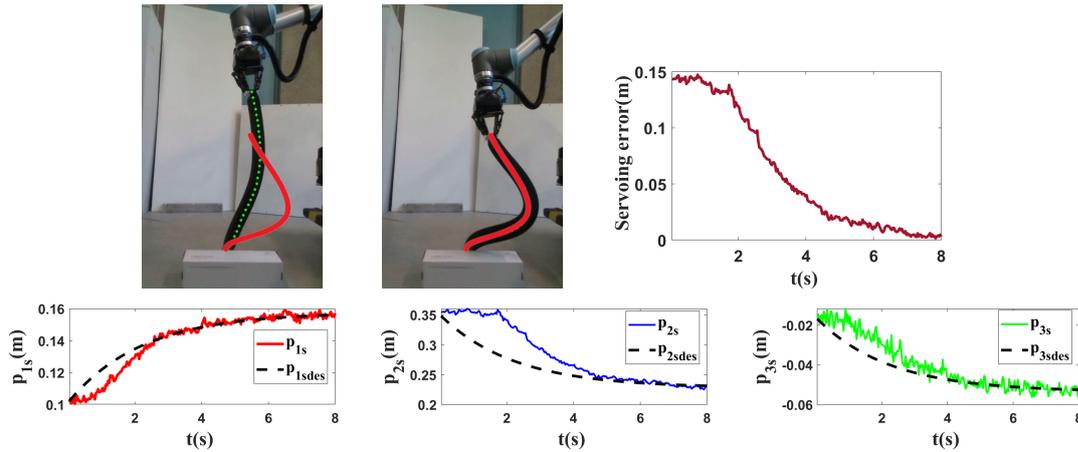


FIGURE 6.7: Experiment 6.1. The used object is a deformable foam rod. From the top-left in the first row, the initial shape, final shape, and servoing error can be seen. The target shape is shown in red. The evolution of the shape (average of its states p_{1ml} , p_{2ml} and p_{3ml}) with respect to its desired trajectory can be found in the second row.

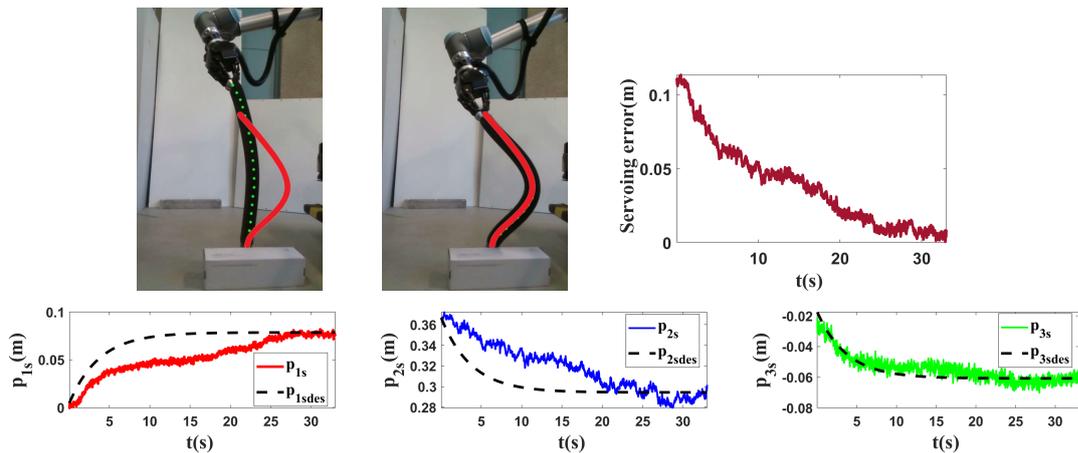


FIGURE 6.8: Experiment 6.2. The used object is a deformable foam rod. From the top-left in the first row, the initial shape, final shape, and servoing error can be seen. The target shape is shown in red. The evolution of the shape (average of its states p_{1ml} , p_{2ml} and p_{3ml}) with respect to its desired trajectory can be found in the second row.

To present the servoing errors, we use the norm of the errors defined from (6.1) as follows:

$$\text{Servoing error} =$$

$$\sqrt{\frac{1}{n^2} \sum_{l=1}^n (p_{1m_l} - p_{1m_lf})^2 + (p_{2m_l} - p_{2m_lf})^2 + (p_{3m_l} - p_{3m_lf})^2} \quad (6.29)$$

In (6.29), the errors of all the points used to represent the object's shape are included. The summation of the errors of these points converging toward zero means that the object reaches its target and the controller objective is completed. Moreover, to show how the object tracks the desired trajectories, we plot the evolution of the states representing the object's shape.

To validate the methodology, we first perform two different experiments on the flexible foam rod by changing the initial shape, target, and the exponential coefficient κ of the desired trajectories. We choose two different values to test the convergence speed of the method. As the results in Fig. 6.7 and Fig. 6.8 illustrate, we can increase or decrease the convergence speed based on the desired trajectories. For instance, in experiment 1 (Fig. 6.7), we increase the exponential coefficient κ to get a quick deformation. However, for the subsequent experiments with natural branches, we decrease κ to avoid causing any breakage since these objects are particularly fragile. The results of all the experiments can be seen in Fig. 6.7 to Fig. 6.12. The initial value, the target value and the final error of p_{1s} , p_{2s} and p_{3s} in the performed experiments can be seen in Table 6.2. We choose different initial and target shapes for each experiment. At each instant, the state of the grasped point is updated using (6.13) and (6.16). The objects' shapes are assumed to remain stable during the entire control process. The servoing errors of these experiments and the evolution of the shape (average of its states p_{1ml} , p_{2ml} and p_{3ml}) regarding its desired trajectory are illustrated in Fig. 6.7 to Fig. 6.12. As can be seen, by applying the controller during the deformation process, the errors converge to zero, and the objects are successfully deformed toward their desired shapes. Note that the deformations achieved are in 3-D space, not being restricted to a fixed plane.

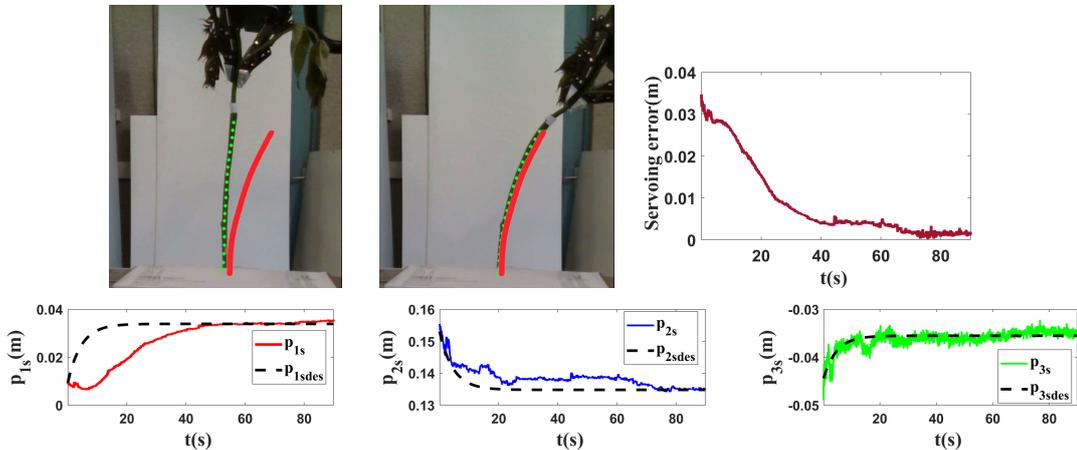


FIGURE 6.9: Experiment 6.3. The used object is a branch of a plant. From the top-left in the first row, the initial shape, final shape, and servoing error are displayed. The target shape is shown in red. The evolution of the shape (average of its states p_{1ml} , p_{2ml} and p_{3ml}) with respect to its desired trajectory can be found in the second row.

6.4.1 Mean squared error (MSE)

Like the previous chapters, the mean squared error (MSE) of all tests is presented in this section. To determine the mean errors between the target and final shape in the performed tests, (3.18) is used to obtain the MSE of each state representing the objects'

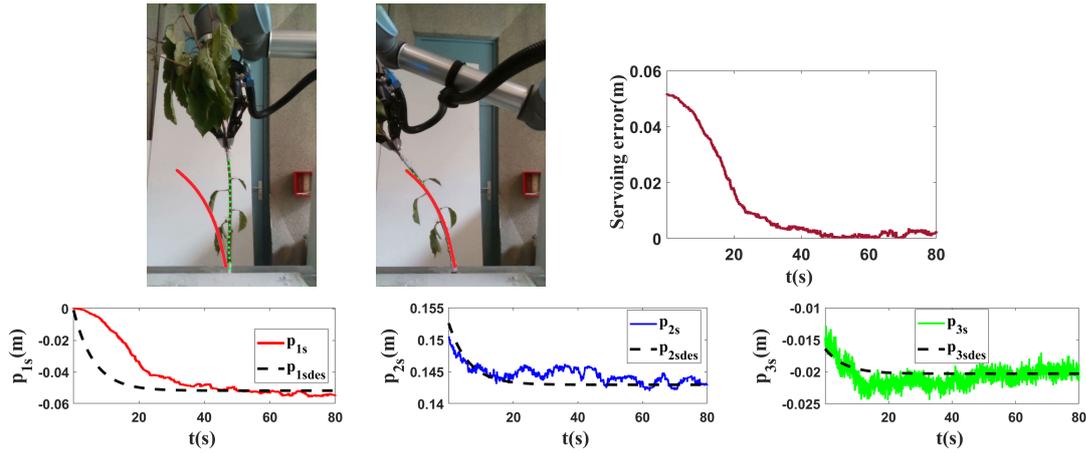


FIGURE 6.10: Experiment 6.4. The used object is a branch of a tree. From the top-left in the first row, the initial shape, final shape, and servoing error are displayed. The target shape is shown in red. The evolution of the shape (average of its states p_{1ml} , p_{2ml} and p_{3ml}) with respect to its desired trajectory can be seen in the second row.

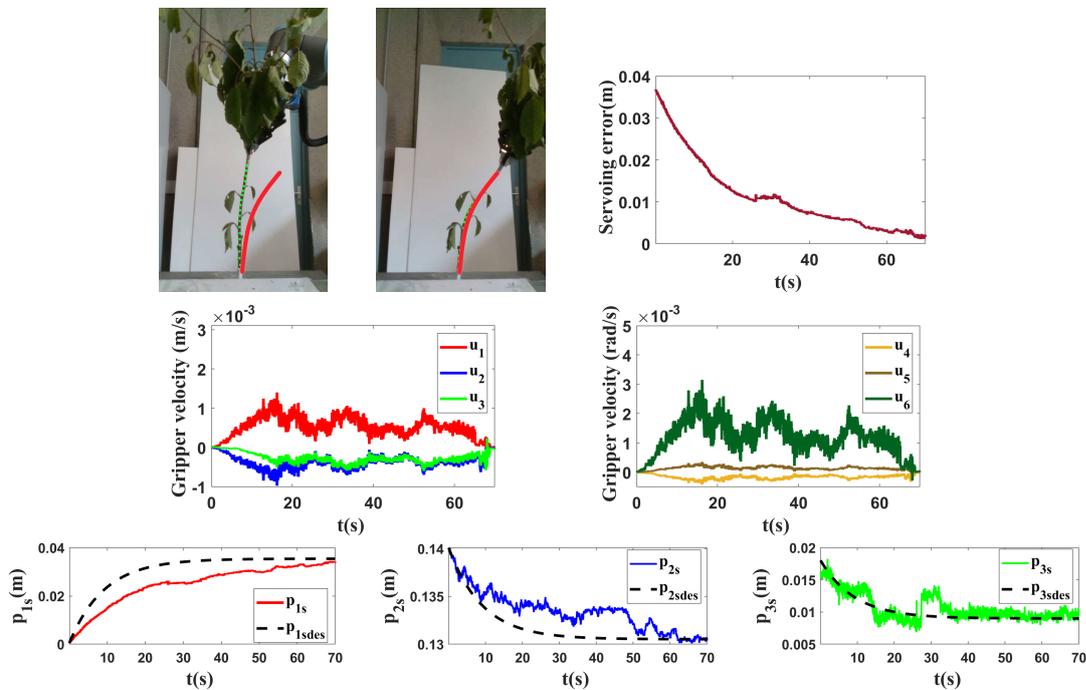


FIGURE 6.11: Experiment 6.5. The used object is a branch of a tree. The initial shape, final shape, and servoing error can be seen in the first row. The target shape is shown in red. The controller inputs (u_1 to u_6) are illustrated in the second row. The evolution of the shape (average of its states p_{1ml} , p_{2ml} and p_{3ml}) with respect to its desired trajectory can be found in the third row.

shape. In this chapter, six different experiments were performed and based on the results presented in Table 6.2, we have:

$$e_{p_1} = 0.005(m), e_{p_2} = 0.005(m), e_{p_3} = 0.004(m) \quad (6.30)$$

As shown, the proposed approach is able to control the shape of the adopted objects despite the various properties of the objects and the existing detection noise.

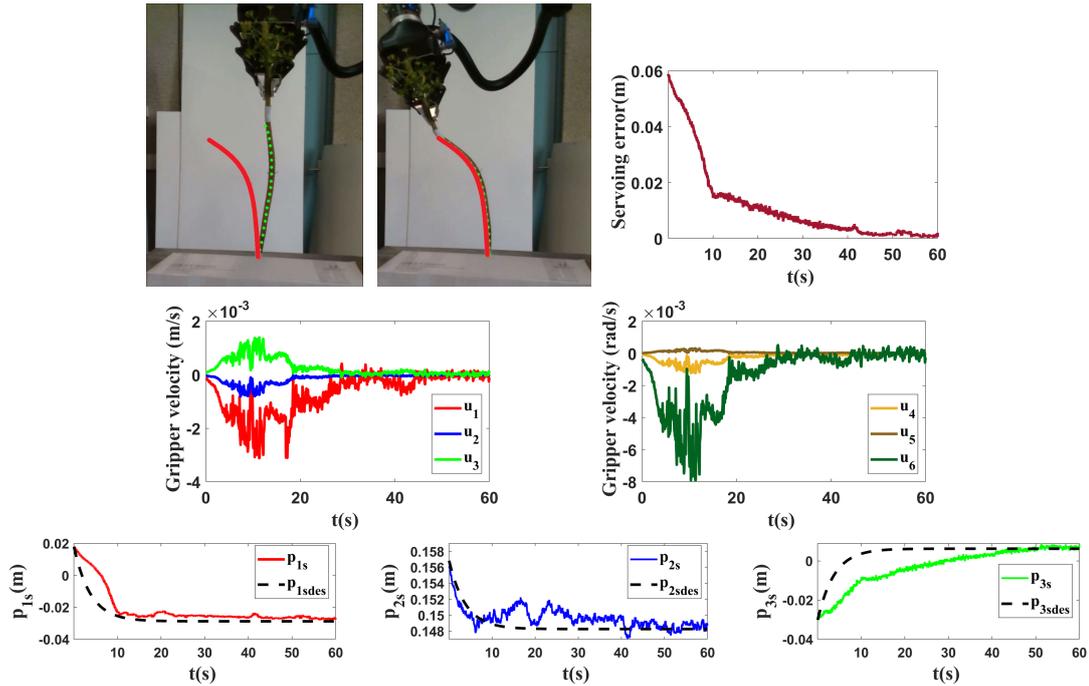


FIGURE 6.12: Experiment 6.6. The used object is a branch of a plant. The initial shape, final shape, and servoing error are indicated in the first row. The target shape is shown in red. The controller inputs (u_i) are presented in the second row. The evolution of the shape (average of its states p_{1ml} , p_{2ml} and p_{3ml}) with respect to its desired trajectory can be found in the third row.

TABLE 6.2: Initial value, target value, and final error of $p_{1s}(m)$, $p_{2s}(m)$, and $p_{3s}(m)$ in the performed tests.

Experiment number	Initial value (m, m, m)	Target value (m, m, m)	Final error (m, m, m)
Exp. 6.1	[0.102, 0.355, -0.016]	[0.156, 0.232, -0.052]	[0.007, 0.003, 0.003]
Exp. 6.2	[0.000, 0.363, -0.021]	[0.079, 0.294, -0.061]	[0.004, 0.009, 0.004]
Exp. 6.3	[0.009, 0.155, -0.046,]	[0.034, 0.135, -0.035]	[0.006, 0.001, 0.003]
Exp. 6.4	[0.000, 0.150, -0.015]	[-0.052, 0.143, -0.020]	[0.003, 0.002, 0.006]
Exp. 6.5	[0.001, 0.140, 0.016]	[0.035, 0.130, 0.009]	[0.001, 0.001, 0.005]
Exp. 6.6	[0.018, 0.156, -0.030]	[-0.029, 0.148, 0.006]	[0.004, 0.006, 0.002]

6.5 Conclusion

In this chapter, we investigated the problem of controlling LDOs in the agricultural field. We presented a method to control the shape of these kinds of objects with a robotic gripper in 3-D. An indirect adaptive control method was developed to reshape the LDOs based on predefined desired shapes. The proposed method is able to track the evolution of the desired trajectories in real time. The method does not require prior knowledge of the object's deformation properties. Furthermore, we are not obliged to calculate the Jacobian matrix based on an offline model to obtain the relationship between the displacements of the manipulator and the deformations of the object.

The proposed method controls the entire shape of the object while following a desired trajectory towards the target shape. The adaptation law to update the estimations of the system parameters and the states of the controlled points during the control process were presented. The validity of the proposed adaptation law was investigated using the Lyapunov theorem and Barbalat's lemma. The controller was successfully validated using

various experiments on a flexible foam rod and branches of various plants, assuming that the object's shape can be tracked.

Chapter 7

Conclusion and perspectives

The contribution of this thesis is on the topic of **DO** robotic manipulation. Several novel approaches to automated deformable shape control using a robotic arm were proposed and experimentally validated. In this chapter, first, the contributions of this dissertation are summarized. Next, the limitations of our proposed methods are reviewed. Following that, the dissertation's future research directions are highlighted.

7.1 Summary

In this dissertation, we have worked on dexterous robotic manipulation of **LDOs**. First, we investigated the existing literature on this topic. We attempted to contribute to the state-of-the-art in robotic manipulation of **DO** with both new algorithms and applications. On the algorithm side, our manipulation schemes take a step in deforming the linear objects without considering the object's material or dynamical parameters such as their stiffness. On the application side, we set a new milestone in manipulating the objects targeting agricultural applications.

In chapter 3, we have presented an approach to control one or several arbitrary points along the length of **LDOs**. A novel method is proposed for **2-D** control of the shape of **LDOs** based on an offline geometric model which, at run-time, does not require computing a deformation model or sensing the entire shape of the object. Our objective was to propose a practical and straightforward method to control **LDOs** based on an offline model that does not need to know any physical or material parameter or characterization of the object's deformation. We only need the knowledge of all the nodes that construct the shape of the object at rest shape so as to compute the offline Jacobian matrix. The developed method requires just a simple offline geometric model (by dividing the object's shape into a set of nodes) based on **ASAP**, which is a significant practical value.

Taking a step further, we have introduced a novel scheme for manipulating an arbitrary point on **LDOs** using an adaptive control scheme targeting agriculture applications in chapter 4. In this chapter, we tried to propose a more applicable method that can be used with various objects without specifying a predefined fixed mesh. Furthermore, our goal was to design a controller that can be used to track the desired dynamic evolution of the states (i.e., desired trajectories), in contrast to existing adaptive approaches that simply reach a fixed set-point without considering any deformation trajectory. This helps to have a closer control on the manipulation of the object and the time of the task's completion. This technique does not require the calculation of the Jacobian to determine the link between the deformation of the object and motion of the gripper. The provided controller also does not require offline information on the deformation of the object. In addition, the stability of the controlled system was investigated using the Lyapunov theory. We presented the adaptation law to update the estimations of the system parameters and the states of the controlled point during the control process. One of the most intriguing

aspects of our method is that it has been applied effectively to a wide range of objects. Even though these two methods (developed in chapters 3 and 4) did not consider the control of the entire shape of the object, controlling the position and orientation at a single point using a grasped point (which can be in a different position from the controlled point) is sufficient for some tasks (such as pruning trees or assembling industrial parts), requires only a single point (not the entire shape of the object) to be perceived online, and implies the system is fully actuated, which can reduce the impact of local minima. The proposed approaches allow deforming elastic linear objects in a controlled way and need only servoing the controlled feature, which is an important practical advantage.

Later in chapters 5 and 6, we investigated a more substantial challenge and tried to control the entire shape of the LDOs, which can be used in various applications such as plant inspection or cable deformation. Like the previous chapters, our objective was to achieve this point without knowing any model parameters of dynamical properties. To reach the stated objective, first, an optimal control method in 2-D using an online Jacobian is provided. Compared to the method presented in chapter 3, the Jacobian calculation of the chapter 5 is more complicated, and the computational time is a little bigger since it is needed to be computed at each time iteration (still, it is fast enough to be used in the online controls). However, this method does not have the limitations of the method presented in chapter 3, such as having fixed-length meshes without sudden deformations. Moreover, this method can be extended to other types of objects (such as 2-D or 3-D shapes) with the same formula. In the first section of this chapter, unlike all the previous works that had adjusted the target shape based on some former tests, an algorithm is proposed to set a desired target shape for the LDO by minimizing the stress along the length of the object. Next, an optimal control strategy based on an online Jacobian was proposed to deform LDOs from initial shapes to the obtained target shapes. To calculate the Jacobian, we used a geometrical model (ARAP) based on the servoing feedback of the LDOs' shape. The proposed algorithm is capable of driving the shape error between the current shape and target to zero, is simple to implement, and does not necessitate collecting the data from a time window to calculate a Jacobian.

Finally, in chapter 6, we have presented an extension of the adaptive control scheme proposed in chapter 4 to control the entire shape of the LDOs in 3-D, particularly by controlling the shape of various plant branches. One of the significant benefits of this strategy is that it is not tied to any specific model or unique object. As a result, it can be utilized with a wide range of objects having varying properties. In the agricultural industry, this is a considerable advantage. This method was designed based on tracking time-varying trajectories instead of fixed distant targets. The advantage is to have a deformation without sudden changes that can be used in large deformations. The proposed method does not require any prior information regarding the object's deformation. In addition, we do not need to compute any offline Jacobian matrix to know how the manipulator's displacements are mapped to the object's deformations. This method is suitable for large deformations. It controls the entire shape of the object and not a simplified representation of it.

It deserves to be stated that all the methods presented in this dissertation work in real-time. The comparison of the provided methods is presented in Appendix A. We believe these works can pave the way for improving the dexterity and the autonomy of robots dealing with unknown DOs in the agricultural field. We hope these works can be a starting point for future research in this field.

7.2 Limitation and future perspective

This thesis makes contributions to the field of vision-based robotic manipulation targeting agricultural applications. In the following, we briefly explore the limitations of the presented work and future lines of research.

The limitations of the approach presented in chapter 3 are that it is restricted to 2-D workspaces, it cannot handle non-smooth unstable deformation behaviors, and the type of objects it can be applied on is limited. This means that we cannot easily extend it into nonlinear objects. Moreover, the method cannot be applied to the objects without the possibility of calculating the Jacobian matrix. The limitations of the method presented in chapter 4 are that it cannot handle non-smooth deformations. It is developed in 2-D; therefore, it is restricted to plane motions. It is not able to do all the tasks at fast speed, and sometimes we need to reduce the convergence speed, which is not easy.

The main limitation of chapter 5 is the proof of global convergence since the system is underactuated (the number of actuated DOF (DOF of the grasped point) is lower than the number of DOF of the object). Another limitation is that the Jacobian calculation is attached to a particular geometrical model (ARAP), so applying it to all types of objects is not always easy. Therefore, this method cannot be used for the objects without the possibility of constructing a geometric model. This method can also control the 2-D shape of the object. The main limitation of the work provided in chapter 6 is that for very complex tasks such as manipulating LDOs to highly curved shapes, our proposed technique (like the methods presented in the previous chapters) may fail since it may be impossible to achieve the target shape only by controlling one end of the LDO, and manipulation of the midpoints of the LDO may be required with more robots.

One of the interesting features of our proposed methods is that we do not consider any force feedback from the object. However, it can be helpful to have force feedback in some tasks that need to measure the force of the gripper. Thus, for future steps, it can be interesting to consider this challenge and propose a method that uses the stress along the object's length as feedback while controlling its shape. This would be helpful to have a complete methodology that ensures no breakage occurs to the studied objects (particularly in manipulating the fragile branches) during the manipulation process. Another interesting future step is optimizing the desired trajectory according to the deformation task. Particularly, our long-term objective for the current adaptive works (i.e., chapters 4 and 6) is to provide a way to define optimal desired trajectories based on the task and object stiffness. We are also interested in exploring a way to use the proposed method while finding suitable grasping points according to the shape control task specifications. It can be interesting to find suitable grasping points for a given shape control task. In chapter 5, we presented a method to design a target shape based on the boundary condition of the objects (i.e., the final pose of its two ends). It could be useful to apply this concept to 3-D shapes in order to build a specific target with the least amount of stress.

Finally, it should be noted that robots' dexterous manipulation of DOs is a promising subject of study with a bright future in agriculture. Hardware advancements will facilitate the creation of new methodologies by providing greater computing capabilities and sensors. There is a wide range of tasks in agriculture that can be considered for future works. In perception, there are still many challenges that can be the research topic of future studies. Although we paved the way in control, applying these methods in the outdoor environment could also be interesting and using these approaches in the field for an outdoor investigation can be quite rewarding. Exploring some perception methods that can detect the object in the actual farm can be helpful.

Despite significant progress in recent years, manipulating DOs still remains an open challenge for robotic manipulators. In this topic, there is still significant opportunities

for novel solutions to be discovered, as well as numerous unsolved challenges. Other potential topics of future research include investigating the constraints identified in this thesis and developing novel strategies to address the unresolved challenges stated above.

Bibliography

- [AACR⁺22] Omid Aghajanzadeh, Miguel Aranda, Juan Antonio Corrales Ramon, Christophe Cariou, Roland Lenain, and Youcef Mezouar. Adaptive deformation control for elastic linear objects. *Frontiers in Robotics and AI*, page 113, 2022.
- [AALN⁺22] Omid Aghajanzadeh, Miguel Aranda, Gonzalo López-Nicolás, Roland Lenain, and Youcef Mezouar. An offline geometric model for controlling the shape of elastic linear objects. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022.
- [ACM19] Miguel Aranda, Juan Antonio Corrales, and Youcef Mezouar. Deformation-based shape control with a multirobot system. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 2174–2180. IEEE, 2019.
- [ACOL00] Marc Alexa, Daniel Cohen-Or, and David Levin. As-rigid-as-possible shape interpolation. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 157–164, 2000.
- [AGDMC16] Yinoussa Adagolodjo, Laurent Goffin, Michel De Mathelin, and Hadrien Courtecuisse. Inverse real-time finite element simulation for robotic control of flexible needle insertion in deformable tissues. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2717–2722. IEEE, 2016.
- [ALNM21] Miguel Aranda, Gonzalo López-Nicolás, and Youcef Mezouar. Distributed linear control of multirobot formations organized in triads. *IEEE Robotics and Automation Letters*, 6(4):8498–8505, Oct. 2021.
- [AM07] Brian DO Anderson and John B Moore. *Optimal control: linear quadratic methods*. Courier Corporation, 2007.
- [APR⁺22] Omid Aghajanzadeh, Guillaume Picard, Juan Antonio Corrales Ramon, Christophe Cariou, Roland Lenain, and 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.
- [ARB⁺21] Arif U Alam, Pranali Rathi, Heba Beshai, Gursimran K Sarabha, and M Jamal Deen. Fruit quality monitoring with smart packaging. *Sensors*, 21(4):1509, 2021.
- [ARM⁺20] Miguel Aranda, Juan Antonio Corrales Ramon, Youcef Mezouar, Adrien Bartoli, and Erol Özgür. Monocular visual shape tracking and servoing for isometrically deforming objects. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7542–7549. IEEE, 2020.

- [ASF20] Omid Aghajanzadeh, Mojtaba Sharifi, and Ali Falsafi. Robust control strategy for hbv treatment: Considering parametric and nonparametric uncertainties. In *Control Applications for Biomedical Engineering Systems*, pages 127–147. Elsevier, 2020.
- [ASTZ17] Omid Aghajanzadeh, Mojtaba Sharifi, Shabnam Tashakori, and Hassan Zohoor. Nonlinear adaptive control method for treatment of uncertain hepatitis b virus infection. *Biomedical Signal Processing and Control*, 38:174–181, 2017.
- [ASTZ18] Omid Aghajanzadeh, Mojtaba Sharifi, Shabnam Tashakori, and Hassan Zohoor. Robust adaptive lyapunov-based control of hepatitis b infection. *IET Systems Biology*, 12(2):62–67, 2018.
- [AWH⁺18a] Farshid Alambeigi, Zerui Wang, Rachel Hegeman, Yun-Hui Liu, and Mehran Armand. Autonomous data-driven manipulation of unknown anisotropic deformable tissues using unmodelled continuum manipulators. *IEEE Robotics and Automation Letters*, 4(2):254–261, 2018.
- [AWH⁺18b] Farshid Alambeigi, Zerui Wang, Rachel Hegeman, Yun-Hui Liu, and Mehran Armand. A robust data-driven approach for online learning and manipulation of unmodeled 3-D heterogeneous compliant objects. *IEEE Robotics and Automation Letters*, 3(4):4140–4147, 2018.
- [AZMD15] Federico Augugliaro, Emanuele Zarfati, Ammar Mirjan, and Raffaello D’Andrea. Knot-tying with flying machines for aerial construction. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5917–5922. IEEE, 2015.
- [Bac15] Shivaji Bachche. Deliberation on design strategies of automatic harvesting systems: A survey. *Robotics*, 4(2):194–222, 2015.
- [BB15] Andy Borum and Timothy Bretl. The free configuration space of a kirchhoff elastic rod is path-connected. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2958–2964. IEEE, 2015.
- [BDB⁺08] Johan Baeten, Kevin Donn e, Sven Boedrij, Wim Beckers, and Eric Claesens. Autonomous fruit picking machine: A robotic apple harvester. In *Field and service robotics*, pages 531–539. Springer, 2008.
- [BDB18] Victor Bloch, Amir Degani, and Avital Bechar. A methodology of orchard architecture design for an optimal harvesting robot. *Biosystems Engineering*, 166:126–137, 2018.
- [Ber13] Dmitry Berenson. Manipulation of deformable objects without modeling and simulating deformation. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4525–4532, 2013.
- [BHW94] David E Breen, Donald H House, and Michael J Wozny. Predicting the drape of woven cloth using interacting particles. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 365–372, 1994.
- [BK19] Aude Billard and Danica Kragic. Trends and challenges in robot manipulation. *Science*, 364(6446):eaat8414, 2019.

- [BLM04] Joel Brown, Jean-Claude Latombe, and Kevin Montgomery. Real-time knot-tying simulation. *The Visual Computer*, 20(2):165–179, 2004.
- [BM14] Timothy Bretl and Zoe McCarthy. Quasi-static manipulation of a kirchhoff elastic rod based on a geometric analysis of equilibrium configurations. *The International Journal of Robotics Research*, 33(1):48–68, 2014.
- [BM17] José María Benítez and Francisco Javier Montáns. The mechanical behavior of skin: Structures and models for the finite element analysis. *Computers & Structures*, 190:75–107, 2017.
- [BMAK14] Jeannette Bohg, Antonio Morales, Tamim Asfour, and Danica Kragic. Data-driven grasp synthesis—a survey. *IEEE Transactions on Robotics*, 30(2):289–309, 2014.
- [BMBS16] Ehud Barnea, Rotem Mairon, and Ohad Ben-Shahar. Colour-agnostic shape-based 3-D fruit detection for crop harvesting robots. *Biosystems Engineering*, 146:57–70, 2016.
- [BMGA10] Jur van den Berg, Stephen Miller, Ken Goldberg, and Pieter Abbeel. Gravity-based robotic cloth folding. In *Algorithmic Foundations of Robotics IX*, pages 409–424. Springer, 2010.
- [BPG⁺17] Tom Botterill, Scott Paulin, Richard Green, Samuel Williams, Jessica Lin, Valerie Saxton, Steven Mills, XiaoQi Chen, and Sam Corbett-Davies. A robot system for pruning grape vines. *Journal of Field Robotics*, 34(6):1100–1122, 2017.
- [BW98] David Baraff and Andrew Witkin. Large steps in cloth simulation. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 43–54, 1998.
- [BWG⁺99] Jeffrey Berkley, Suzanne Weghorst, Hayes Gladstone, Gregory Raugi, Daniel Berg, and Mark Ganter. Fast finite element modeling for surgical simulation. *Studies in health technology and informatics*, pages 55–61, 1999.
- [CCR21] Stéphane Caro, Christine Chevallereau, and Alberto Remus. Manipulating deformable objects with a dual-arm robot. In *ROBOVIS 2021: 2nd International Conference on Robotics, Computer Vision and Intelligent Systems*, 2021.
- [CFL⁺20] Konstantinos Chatzilygeroudis, Bernardo Fichera, Ilaria Lauzana, Fanjun Bu, Kunpeng Yao, Farshad Khadivar, and Aude Billard. Benchmark for bimanual robotic manipulation of semi-deformable objects. *IEEE Robotics and Automation Letters*, 5(2):2443–2450, 2020.
- [CG91] George Celniker and Dave Gossard. Deformable curve and surface finite-elements for free-form shape design. In *Proceedings of the 18th annual conference on Computer graphics and interactive techniques*, pages 257–266, 1991.
- [CG16] Renjie Chen and Craig Gotsman. Generalized as-similar-as-possible warping with applications in digital photography. *Computer Graphics Forum*, 35(2):81–92, 2016.

- [CGLX17] Shu-Yu Chen, Lin Gao, Yu-Kun Lai, and Shihong Xia. Rigidity controllable as-rigid-as-possible shape deformation. *Graphical Models*, 91:13–21, 2017.
- [Cha04] François Chaumette. Image moments: a general and useful set of features for visual servoing. *IEEE Transactions on Robotics*, 20(4):713–723, 2004.
- [CHP89] John E Chadwick, David R Haumann, and Richard E Parent. Layered construction for deformable animated characters. *ACM Siggraph Computer Graphics*, 23(3):243–252, 1989.
- [CLP⁺20] Lin Cao, Xiaoguo Li, Phuoc Thien Phan, Anthony Meng Huat Tiong, Hung Leng Kaan, Jiajun Liu, Wenjie Lai, Yanpei Huang, Huu Minh Le, Muneaki Miyasaka, et al. Sewing up the wounds: A robotic suturing system for flexible endoscopy. *IEEE Robotics & Automation Magazine*, 27(3):45–54, 2020.
- [COC⁺20] Andrea Cherubini, Valerio Ortenzi, Akansel Cosgun, Robert Lee, and Peter Corke. Model-free vision-based shaping of deformable plastic materials. *The International Journal of Robotics Research*, 39(14):1739–1759, 2020.
- [CP20] Peng Chang and Taşkın Padır. Model-based manipulation of linear flexible objects with visual curvature feedback. In *2020 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 1406–1412. IEEE, 2020.
- [CZ92] David T Chen and David Zeltzer. Pump it up: Computer animation of a biomechanically based model of muscle using the finite element method. In *Proceedings of the 19th annual conference on computer graphics and interactive techniques*, pages 89–98, 1992.
- [Das10] Jadav Das. *Robotic control of deformable continua and objects therein*. Vanderbilt University, 2010.
- [Dav06] Thomas L Davenport. Pruning strategies to maximize tropical mango production from the time of planting to restoration of old orchards. *HortScience*, 41(3):544–548, 2006.
- [DBM⁺20] Joseph Davidson, Santosh Bhusal, Changki Mo, Manoj Karkee, and Qin Zhang. Robotic manipulation for specialty crop harvesting: A review of manipulator and end-effector technologies. *Global Journal of Agricultural and Allied Sciences*, 2(1):25–41, 2020.
- [DBPC18] Simon Duenser, James M Bern, Roi Poranne, and 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.
- [DDF⁺17] Mingsong Dou, Philip Davidson, Sean Ryan Fanello, Sameh Khamis, Adarsh Kowdle, Christoph Rhemann, Vladimir Tankovich, and Shahram Izadi. Motion2fusion: Real-time volumetric performance capture. *ACM Transactions on Graphics (TOG)*, 36(6):1–16, 2017.

- [DHYH18] Feng Ding, Jian Huang, Wei Yao, and Shunfan He. Anti-swing control in manipulation of a deformable linear object using dynamic surface control. In *2018 3rd International Conference on Advanced Robotics and Mechatronics (ICARM)*, pages 503–508. IEEE, 2018.
- [DRCRM⁺17] Ángel Delgado Rodríguez, Juan Antonio Corrales Ramón, Youcef Mezouar, Laurent Lequievre, Carlos A Jara, and Fernando Torres. Tactile control based on gaussian images and its application in bi-manual manipulation of deformable objects. 2017.
- [DS11] Jadav Das and Nilanjan Sarkar. Autonomous shape control of a deformable object by multiple manipulators. *Journal of Intelligent & Robotic Systems*, 62(1):3–27, 2011.
- [EKS03] Olaf Etzmuß, Michael Keckeisen, and Wolfgang Straßer. A fast finite element solution for cloth modelling. In *11th Pacific Conference on Computer Graphics and Applications, 2003. Proceedings.*, pages 244–251. IEEE, 2003.
- [FMC⁺18] Fanny Ficuciello, Alessandro Migliozi, Eulalie Coevoet, Antoine Petit, and Christian Duriez. Fem-based deformation control for dexterous manipulation of 3-D soft objects. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4007–4013. IEEE, 2018.
- [FP04] Gian Luca Foresti and Felice Andrea Pellegrino. Automatic visual recognition of deformable objects for grasping and manipulation. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 34(3):325–333, 2004.
- [FSAB11] Barbara Frank, Cyrill Stachniss, Nichola Abdo, and Wolfram Burgard. Efficient motion planning for manipulation robots in environments with deformable objects. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2180–2185. IEEE, 2011.
- [FSTB21] Paula Fernández-Serrano, Paula Tarancón, and Cristina Besada. Consumer information needs and sensory label design for fresh fruit packaging. an exploratory study in spain. *Foods*, 10(1):72, 2021.
- [FZF⁺18] Qingchun Feng, Wei Zou, Pengfei Fan, Chunfeng Zhang, and Xiu Wang. Design and test of robotic harvesting system for cherry tomato. *International Journal of Agricultural and Biological Engineering*, 11(1):96–100, 2018.
- [GJMSMCMJ14] Sergio Garrido-Jurado, Rafael Muñoz-Salinas, Francisco José Madrid-Cuevas, and Manuel Jesús Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6):2280–2292, 2014.
- [GN03] Michael A Greminger and Bradley J Nelson. Deformable object tracking using the boundary element method. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, volume 1, pages I–I. IEEE, 2003.

- [GSN04] Michael A Greminger, Yu Sun, and Bradley J Nelson. Boundary element deformable object tracking with equilibrium constraints. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, volume 4, pages 3896–3901. IEEE, 2004.
- [HA94] Koh Hosoda and Minoru Asada. Versatile visual servoing without knowledge of true jacobian. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'94)*, volume 1, pages 186–193. IEEE, 1994.
- [HBvT⁺14] J Hemming, CW Bac, BAJ van Tuijl, R Barth, J Bontsema, EJ Pekkeriet, and E van Henten. A robot for harvesting sweet-pepper in greenhouses. In *AgEng 2014*, 2014.
- [HDL⁺21] Shengzeng Huo, Anqing Duan, Chengxi Li, Peng Zhou, Wanyu Ma, and David Navarro-Alarcon. Keypoint-based bimanual shaping of deformable linear objects under environmental constraints using hierarchical action planning. *arXiv preprint arXiv:2110.08962*, 2021.
- [HHS⁺19] Zhe Hu, Tao Han, Peigen Sun, Jia Pan, and Dinesh Manocha. 3-D deformable object manipulation using deep neural networks. *IEEE Robotics and Automation Letters*, 4(4):4255–4261, 2019.
- [HKL⁺12] Kil-Su Han, Si-Chan Kim, Young-Bum Lee, Sang-Chul Kim, Dong-Hyuk Im, Hong-Ki Choi, and Heon Hwang. Strawberry harvesting robot for bench-type cultivation. *Journal of Biosystems Engineering*, 37(1):65–74, 2012.
- [HLNAS19] Rafael Herguedas, Gonzalo López-Nicolás, Rosario Aragüés, and 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.
- [HS18] Long He and James Schupp. Sensing and automation in pruning of apple trees: A review. *Agronomy*, 8(10):211, 2018.
- [HSP18] Zhe Hu, Peigen Sun, and Jia Pan. Three-dimensional deformable object manipulation using fast online gaussian process regression. *IEEE Robotics and Automation Letters*, 3(2):979–986, 2018.
- [HTW01] Shinichi Hirai, Tatsuhiko Tsuboi, and 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.
- [HTY05] Jian Huang, Isao Todo, and Tetsuro Yabuta. Position/force hybrid control of a manipulator with a flexible tool using visual and force information. In *Cutting Edge Robotics*. Citeseer, 2005.
- [HYK10] Mitsuru Higashimori, Kayo Yoshimoto, and Makoto Kaneko. Active shaping of an unknown rheological object based on deformation decomposition into elasticity and plasticity. In *2010 IEEE International Conference on Robotics and Automation*, pages 5120–5126. IEEE, 2010.

- [HZSP18] Tao Han, Xuan Zhao, Peigen Sun, and Jia Pan. Robust shape estimation for 3-D deformable object manipulation. *Communications in Information and Systems*, 18(2):107–124, 2018.
- [II84] H Inoue and M Inaba. Hand-eye coordination in rope handling, robotics research: The first international symposium, 1984.
- [IMH05] Takeo Igarashi, Tomer Moscovich, and John F. Hughes. As-rigid-as-possible shape manipulation. *ACM Trans. Graph.*, 24(3):1134–1141, 2005.
- [JC07] Hongjun Jeon and Min-Hyung Choi. Interactive motion control of deformable objects using localized optimal control. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 2582–2587. IEEE, 2007.
- [JÇ13] Russell C Jackson and M Cenk Çavuşoğlu. Needle path planning for autonomous robotic surgical suturing. In *2013 IEEE International Conference on Robotics and Automation*, pages 1669–1675. IEEE, 2013.
- [JL97] Ammar Joukhadar and Christian Laugier. Dynamic simulation: Model, basic algorithms, and optimization. *Algorithms For Robotic Motion and Manipulation*, pages 419–434, 1997.
- [JP99] Doug L James and Dinesh K Pai. Artdefo: accurate real time deformable objects. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 65–72, 1999.
- [JP03] Doug L James and Dinesh K Pai. Multiresolution green’s function methods for interactive simulation of large-scale elastostatic objects. *ACM Transactions on Graphics (TOG)*, 22(1):47–82, 2003.
- [JWT19] Shiyu Jin, Changhao Wang, and Masayoshi Tomizuka. Robust deformation model approximation for robotic cable manipulation. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6586–6593. IEEE, 2019.
- [KFB⁺21] A Koessler, N Roca Filella, BC Bouzgarrou, L Lequievre, and 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.
- [KGKG98] Erwin Keeve, Sabine Girod, Ron Kikinis, and Bernd Girod. Deformable modeling of facial tissue for craniofacial surgery simulation. *Computer Aided Surgery: Official Journal of the International Society for Computer Aided Surgery (ISCAS)*, 3(5):228–238, 1998.
- [KP12] Steven Kinio and Alexandru Patriciu. A comparative study of H_∞ and PID control for indirect deformable object manipulation. In *2012 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 414–420. IEEE, 2012.
- [KSLO96] Lydia E Kavraki, Petr Svestka, J-C Latombe, and Mark H Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE transactions on Robotics and Automation*, 12(4):566–580, 1996.

- [KW02] Hyosig Kang and John T Wen. Robotic knot tying in minimally invasive surgeries. In *IEEE/RSJ International conference on intelligent robots and systems*, volume 2, pages 1421–1426. IEEE, 2002.
- [KYS⁺09] Naoshi Kondo, Kazuya Yamamoto, Hiroshi Shimizu, Koki Yata, Mitsutaka Kurita, Tomoo Shiigi, Mitsuji Monta, and Takahisa Nishizu. A machine vision system for tomato cluster harvesting robot. *Engineering in Agriculture, Environment and Food*, 2(2):60–65, 2009.
- [LAAB14] Ibai Leizea, Hugo Alvarez, Iker Aguinaga, and Diego Borro. Real-time deformation, registration and tracking of solids based on physical simulation. In *2014 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 165–170. IEEE, 2014.
- [LG14] Zohar Levi and Craig Gotsman. Smooth rotation enhanced as-rigid-as-possible mesh animation. *IEEE transactions on visualization and computer graphics*, 21(2):264–277, 2014.
- [LGWJ15] Huan Lin, Feng Guo, Feifei Wang, and Yan-Bin Jia. Picking up a soft 3-D object by “feeling” the grip. *The International Journal of Robotics Research*, 34(11):1361–1384, 2015.
- [LH06] Xiadong Lv and Xinhan Huang. Fuzzy adaptive kalman filtering based estimation of image jacobian for uncalibrated visual servoing. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2167–2172. IEEE, 2006.
- [LK01] Florent Lamiraux and Lydia E Kavraki. Planning paths for elastic objects under manipulation constraints. *The International Journal of Robotics Research*, 20(3):188–208, 2001.
- [LK21] Rita Laezza and Yiannis Karayiannidis. Learning shape control of elastoplastic deformable linear objects. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4438–4444, 2021.
- [LKM20] Romain Lagneau, Alexandre Krupa, and Maud Marchal. Automatic shape control of deformable wires based on model-free visual servoing. *IEEE Robotics and Automation Letters*, 5(4):5252–5259, 2020.
- [LL08] Honghai Liu and Hua Lin. Sequence trajectory generation for garment handling systems. *Applied mathematical modelling*, 32(6):1017–1026, 2008.
- [LLJ22] Naijing Lv, Jianhua Liu, and Yunyi Jia. Dynamic modeling and control of deformable linear objects for single-arm and dual-arm robot manipulations. *IEEE Transactions on Robotics*, 2022.
- [LSGL18] Xiang Li, Xing Su, Yuan Gao, and Yun-Hui Liu. Vision-based robotic grasping and manipulation of usb wires. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3482–3487. IEEE, 2018.
- [Lue97] David G Luenberger. *Optimization by vector space methods*. John Wiley & Sons, 1997.
- [LWL19] Xiang Li, Zerui Wang, and Yun-Hui Liu. Sequential robotic manipulation for active shape control of deformable linear objects. In *2019 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, pages 840–845. IEEE, 2019.

- [LZX⁺08] Ligang Liu, Lei Zhang, Yin Xu, Craig Gotsman, and Steven J. Gortler. A local/global approach to mesh parameterization. In *Proceedings of the Symposium on Geometry Processing*, pages 1495–1504, 2008.
- [MB18] Dale McConachie and Dmitry Berenson. Estimating model utility for deformable object manipulation using multiarmed bandit methods. *IEEE Transactions on Automation Science and Engineering*, 15(3):967–979, 2018.
- [MB20] Dale McConachie and Dmitry Berenson. Bandit-based model selection for deformable object manipulation. In *Algorithmic Foundations of Robotics XII*, pages 704–719. Springer, 2020.
- [MHHR07a] Matthias Müller, Bruno Heidelberger, Marcus Hennix, and John Ratcliff. Position based dynamics. *Journal of Visual Communication and Image Representation*, 18(2):109–118, 2007.
- [MHHR07b] Matthias Müller, Bruno Heidelberger, Marcus Hennix, and John Ratcliff. Position based dynamics. *Journal of Visual Communication and Image Representation*, 18(2):109–118, 2007.
- [MK06] Mark Moll and Lydia E Kavraki. Path planning for deformable linear objects. *IEEE Transactions on Robotics*, 22(4):625–636, 2006.
- [MM07] Patricia Moore and Derek Molloy. A survey of computer-based deformable models. In *International Machine Vision and Image Processing Conference (IMVIP 2007)*, pages 55–66. IEEE, 2007.
- [MMA⁺01] Carlos Monserrat, Ullrich Meier, M Alcaniz, Francisco Chinesta, and Mari C Juan. A new approach for the real-time simulation of tissue deformations in surgery simulation. *Computer Methods and Programs in Biomedicine*, 64(2):77–85, 2001.
- [MVDBF⁺12] Stephen Miller, Jur Van Den Berg, Mario Fritz, Trevor Darrell, Ken Goldberg, and Pieter Abbeel. A geometric approach to robotic laundry folding. *The International Journal of Robotics Research*, 31(2):249–267, 2012.
- [NAL13] David Navarro-Alarcon and Yun-hui Liu. Uncalibrated vision-based deformation control of compliant objects with online estimation of the jacobian matrix. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4977–4982. IEEE, 2013.
- [NAL14] David Navarro-Alarcon and Yun-hui Liu. A dynamic and uncalibrated method to visually servo-control elastic deformations by fully-constrained robotic grippers. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4457–4462. IEEE, 2014.
- [NAL17] David Navarro-Alarcon and 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*, 34(1):272–279, 2017.
- [NALRL13a] D. Navarro-Alarcon, Y.-H. Liu, J.G. Romero, and P. Li. Model-free Visually Servoed Deformation Control of Elastic Objects by Robot Manipulators. *IEEE Transactions on Robotics*, 29(6):1457–1468, 2013.

- [NALRL13b] David Navarro-Alarcon, Yunhui Liu, José Guadalupe Romero, and Peng Li. Visually servoed deformation control by robot manipulators. In *2013 IEEE International Conference on Robotics and Automation*, pages 5259–5264. IEEE, 2013.
- [NALRL14] David Navarro-Alarcon, Yun-hui Liu, Jose Guadalupe Romero, and Peng Li. On the visual deformation servoing of compliant objects: Uncalibrated control methods and experiments. *The International Journal of Robotics Research*, 33(11):1462–1480, 2014.
- [NAYW⁺16] David Navarro-Alarcon, Hiu Man Yip, Zerui Wang, Yun-Hui Liu, Fangxun Zhong, Tianxue Zhang, and Peng Li. Automatic 3-D manipulation of soft objects by robotic arms with an adaptive deformation model. *IEEE Transactions on Robotics*, 32(2):429–441, 2016.
- [NCA⁺17] Ashvin Nair, Dian Chen, Pulkit Agrawal, Phillip Isola, Pieter Abbeel, Jitendra Malik, and Sergey Levine. Combining self-supervised learning and imitation for vision-based rope manipulation. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 2146–2153. IEEE, 2017.
- [NP19] Félix Nadon and Pierre Payeur. Automatic selection of grasping points for shape control of non-rigid objects. In *2019 IEEE International Symposium on Robotic and Sensors Environments (ROSE)*, pages 1–7. IEEE, 2019.
- [NP20] Félix Nadon and Pierre Payeur. Grasp selection for in-hand robotic manipulation of non-rigid objects with shape control. In *2020 IEEE International Systems Conference (SysCon)*, pages 1–8. IEEE, 2020.
- [Ogd97] Raymond W Ogden. *Non-linear elastic deformations*. Courier Corporation, 1997.
- [P⁺11] Sachin Patil et al. Motion planning under uncertainty in highly deformable environments. *Robotics science and systems: online proceedings*, 2011.
- [PBG62] LS Pontryagin, VG Boltyanskii, RV Gamkrelidze, and EF Mishchenko. The maximum principle. *The Mathematical Theory of Optimal Processes*. New York: John Wiley and Sons, 1962.
- [PLK02] Jeff Phillips, Andrew Ladd, and Lydia E Kavraki. Simulated knot tying. In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, volume 1, pages 841–846. IEEE, 2002.
- [PLS15] Antoine Petit, Vincenzo Lippiello, and Bruno Siciliano. Real-time tracking of 3-D elastic objects with an RGB-D sensor. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3914–3921. IEEE, 2015.
- [PML04] Jenelle Armstrong Piepmeier, Gary V McMurray, and Harvey Lipkin. Uncalibrated dynamic visual servoing. *IEEE Transactions on Robotics and Automation*, 20(1):143–147, 2004.
- [PMY⁺22] J.A. Preiss, D. Millard, T. Yao, , and G.S. Sukhatme. Tracking fast trajectories with a deformable object using a learned model. In *2022 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2022.

- [PPBC15] Shaifali Parashar, Daniel Pizarro, Adrien Bartoli, and Toby Collins. As-rigid-as-possible volumetric shape-from-template. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 891–899, 2015.
- [QMNA⁺20] Jiaming Qi, Wanyu Ma, David Navarro-Alarcon, Han Gao, and Guangfu Ma. Adaptive shape servoing of elastic rods using parameterized regression features and auto-tuning motion controls. *arXiv preprint arXiv:2008.06896*, 2020.
- [QMZ⁺21] Jiaming Qi, Guangfu Ma, Jihong Zhu, Peng Zhou, Yueyong Lyu, Haibo Zhang, and David Navarro-Alarcon. Contour moments based manipulation of composite rigid-deformable objects with finite time model estimation and shape/position control. *IEEE/ASME Trans. Mechatr.*, pages 1–12, 2021.
- [RMB18] Mengyao Ruan, Dale McConachie, and Dmitry Berenson. Accounting for directional rigidity and constraints in control for manipulation of deformable objects without physical simulation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 512–519. IEEE, 2018.
- [RSHP⁺18] Redmond R Shamshiri, Ibrahim A Hameed, Lenka Pitonakova, Cornelia Weltzien, Siva K Balasundram, Ian J Yule, Tony E Grift, and Girish Chowdhary. Simulation software and virtual environments for acceleration of agricultural robotics: Features highlights and performance comparison. 2018.
- [SA07] Olga Sorkine and Marc Alexa. As-rigid-as-possible surface modeling. In *Symposium on Geometry processing*, volume 4, pages 109–116, 2007.
- [SBAMO22] Mohammadreza Shetab-Bushehri, Miguel Aranda, Youcef Mezouar, and Erol Ozgur. As-rigid-as-possible shape servoing. *IEEE Robotics and Automation Letters*, 7(2):3898–3905, 2022.
- [SCBM18] Jose Sanchez, Juan-Antonio Corrales, Belhassen-Chedli Bouzgarrou, and Youcef Mezouar. Robotic manipulation and sensing of deformable objects in domestic and industrial applications: a survey. *The International Journal of Robotics Research*, 37(7):688–716, 2018.
- [SCOL⁺04] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, and H.-P. Seidel. Laplacian surface editing. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, pages 175–184, 2004.
- [SH06] Mizuho Shibata and Shinichi Hirai. Soft object manipulation by simultaneous control of motion and deformation. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 2460–2465. IEEE, 2006.
- [SHKW18] Redmond R Shamshiri, Ibrahim A Hameed, Manoj Karkee, and Cornelia Weltzien. Robotic harvesting of fruiting vegetables: A simulation approach in v-rep, ros and matlab. *Proceedings in Automation in Agriculture-Securing Food Supplies for Future Generations*, 126, 2018.

- [SHLA16] John Schulman, Jonathan Ho, Cameron Lee, and Pieter Abbeel. Learning from demonstrations through the use of non-rigid registration. In *Robotics Research*, pages 339–354. Springer, 2016.
- [SIL08] Mitul Saha, Pekka Isto, and Jean-Claude Latombe. Motion planning for robotic manipulation of deformable linear objects. In *Experimental Robotics*, pages 23–32. Springer, 2008.
- [SKM21] Agniva Sengupta, Alexandre Krupa, and Eric Marchand. Visual tracking of deforming objects using physics-based models. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 14178–14184. IEEE, 2021.
- [SL⁺91] Jean-Jacques E Slotine, Weiping Li, et al. *Applied nonlinear control*, volume 199. Prentice hall Englewood Cliffs, NJ, 1991.
- [SMBB20] Avishai Sintov, Steven Macenski, Andy Borum, and Timothy Bretl. Motion planning for dual-arm manipulation of elastic rods. *IEEE Robotics and Automation Letters*, 5(4):6065–6072, 2020.
- [SMEDC⁺20] José Sanchez, Kamal Mohy El Dine, Juan Antonio Corrales, Belhassen-Chedli Bouzgarrou, and Youcef Mezouar. Blind manipulation of deformable objects based on force sensing and finite element modeling. *Frontiers in Robotics and AI*, 7:73, 2020.
- [SN02] Mozafar Saadat and Ping Nan. Industrial applications of automatic manipulation of flexible materials. *Industrial Robot: An International Journal*, 2002.
- [SNR96] Philip W Smith, Nagaraj Nandhakumar, and Arvind K Ramadorai. Vision based manipulation of non-rigid objects. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 4, pages 3191–3196. IEEE, 1996.
- [SP86] Thomas W Sederberg and Scott R Parry. Free-form deformation of solid geometric models. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pages 151–160, 1986.
- [SP09] Jerzy Smolen and Alexandru Patriciu. Deformation planning for robotic soft tissue manipulation. In *2009 Second International Conferences on Advances in Computer-Human Interactions*, pages 199–204. IEEE, 2009.
- [SS16] Ankit J Shah and Julie A Shah. Towards manipulation planning for multiple interlinked deformable linear objects. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3908–3915. IEEE, 2016.
- [SSM21] Gustavo Scalabrini Sampaio, Leandro Augusto da Silva, and Maurício Marengoni. 3-D reconstruction of non-rigid plants and sensor data fusion for agriculture phenotyping. *Sensors*, 21(12):4115, 2021.
- [Tan06] Herbert G Tanner. Mobile manipulation of flexible objects under deformation constraints. *IEEE transactions on robotics*, 22(1):179–184, 2006.

- [TFH99] Shinichi Tokumoto, Yoshiaki Fujita, and Shinichi Hirai. Deformation modeling of viscoelastic objects for their shape control. In *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C)*, volume 1, pages 767–772. IEEE, 1999.
- [TKK01] HG Tanner, KJ Kyriakopoulos, and NI Krikelis. Advanced agricultural robots: kinematics and dynamics of multiple mobile manipulators handling non-rigid material. *Computers and electronics in agriculture*, 31(1):91–105, 2001.
- [TM10] Omar Tahri and Youcef Mezouar. On visual servoing based on efficient second order minimization. *Robotics and Autonomous Systems*, 58(5):712–719, 2010.
- [TPBF87] Demetri Terzopoulos, John Platt, Alan Barr, and Kurt Fleischer. Elastically deformable models. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 205–214, 1987.
- [TT94] Xiaoyuan Tu and Demetri Terzopoulos. Artificial fishes: Physics, locomotion, perception, behavior. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 43–50, 1994.
- [TW90] Demetri Terzopoulos and Keith Waters. Physically-based facial modelling, analysis, and animation. *The journal of visualization and computer animation*, 1(2):73–80, 1990.
- [TWT18] Te Tang, Changhao Wang, and Masayoshi Tomizuka. A framework for manipulating deformable linear objects by coherent point drift. *IEEE Robotics and Automation Letters*, 3(4):3426–3433, 2018.
- [Wat87] Keith Waters. A muscle model for animation three-dimensional facial expression. *Acm siggraph computer graphics*, 21(4):17–24, 1987.
- [WBB15] Weifu Wang, Dmitry Berenson, and Devin Balkcom. An online method for tight-tolerance insertion tasks for string and rope. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2488–2495. IEEE, 2015.
- [WH04] Hidefumi Wakamatsu and Shinichi Hirai. Static modeling of linear object deformation based on differential geometry. *The International Journal of Robotics Research*, 23(3):293–311, 2004.
- [WYW⁺18] Hesheng Wang, Bohan Yang, Jingchuan Wang, Xinwu Liang, Weidong Chen, and Yun-Hui Liu. Adaptive visual servoing of contour features. *IEEE/ASME Transactions on Mechatronics*, 23(2):811–822, 2018.
- [YGL⁺18] Jie Yang, Lin Gao, Yu-Kun Lai, Paul L Rosin, and Shihong Xia. Biharmonic deformation transfer with automatic key point selection. *Graphical Models*, 98:1–13, 2018.
- [YGSD22] Alexander You, Cindy Grimm, Abhisesh Silwal, and Joseph R Davidson. Semantics-guided skeletonization of upright fruiting offshoot trees for robotic pruning. *Computers and Electronics in Agriculture*, 192:106622, 2022.

- [YLZL22] Mingrui Yu, Kangchen Lv, Hanzhong Zhong, and Xiang Li. Global model learning for large deformation control of elastic deformable linear objects: An efficient and adaptive approach. *arXiv preprint arXiv:2205.04004*, 2022.
- [YPS⁺21] Francisco Yandun, Tanvir Parhar, Abhisesh Silwal, David Clifford, Zhiqiang Yuan, Gabriella Levine, Sergey Yaroshenko, and George Kantor. Reaching pruning locations in a vine using a deep reinforcement learning policy. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2400–2406. IEEE, 2021.
- [YSF⁺20] Alexander You, Fouad Sukkar, Robert Fitch, Manoj Karkee, and Joseph R Davidson. An efficient planning and control framework for pruning fruit trees. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3930–3936. IEEE, 2020.
- [YVK21] Hang Yin, Anastasia Varava, and Danica Kragic. Modeling, learning, perception, and control methods for deformable object manipulation. volume 6, page eabd8803. American Association for the Advancement of Science, 2021.
- [YZJB20] Mengyuan Yan, Yilin Zhu, Ning Jin, and Jeannette Bohg. Self-supervised learning of state estimation for manipulating deformable linear objects. *IEEE Robotics and Automation Letters*, 5(2):2372–2379, 2020.
- [YZL22] Mingrui Yu, Hanzhong Zhong, and Xiang Li. Shape control of deformable linear objects with offline and online learning of local linear deformation models. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 1337–1343, 2022.
- [YZZL21] Mingrui Yu, Hanzhong Zhong, Fangxun Zhong, and Xiang Li. Adaptive control for robotic manipulation of deformable linear objects with offline and online learning of unknown models. *arXiv preprint arXiv:2107.00194*, 2021.
- [ZBD⁺17] Zhongkai Zhang, Thor Morales Bieze, Jérémie Dequidt, Alexandre Kruszewski, and 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.
- [ZCB⁺17] Lazher Zaidi, Juan Antonio Corrales, Belhassen Chedli Bouzgarrou, Youcef Mezouar, and Laurent Sabourin. Model-based strategy for grasping 3-D deformable objects using a multi-fingered robotic hand. *Robotics and Autonomous Systems*, 95:196–206, 2017.
- [ZCD⁺21] Jihong Zhu, Andrea Cherubini, Claire Dune, David Navarro-Alarcon, Farshid Alambeigi, Dmitry Berenson, Fanny Ficuciello, Kensuke Harada, Xiang Li, Jia Pan, et al. Challenges and outlook in robotic manipulation of deformable objects. *arXiv preprint arXiv:2105.01767*, 2021.
- [ZGT⁺18] Baohua Zhang, Baoxing Gu, Guangzhao Tian, Jun Zhou, Jichao Huang, and Yingjun Xiong. Challenges and solutions of optical-based nondestructive quality inspection for robotic fruit and vegetable grading systems: A technical review. *Trends in food science & technology*, 81:213–231, 2018.

- [ZNAPC21] Jihong Zhu, David Navarro-Alarcon, Robin Passama, and Andrea Cherubini. Vision-based manipulation of deformable and rigid objects using subspace projections of 2-D contours. *Robotics and Autonomous Systems*, 142:103798, 2021.
- [ZNF⁺18] Jihong Zhu, Benjamin Navarro, Philippe Fraitse, André Crosnier, and 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.
- [ZNP⁺19] Jihong Zhu, Benjamin Navarro, Robin Passama, Philippe Fraitse, André Crosnier, and Andrea Cherubini. Robotic manipulation planning for shaping deformable linear objects with environmental contacts. *IEEE Robotics and Automation Letters*, 5(1):16–23, 2019.
- [ZPC21] Simon Zimmermann, Roi Poranne, and Stelian Coros. Dynamic manipulation of deformable objects with implicit integration. *IEEE Robotics and Automation Letters*, 6(2):4209–4216, 2021.
- [ZSGS12] Michael Zollhöfer, Ezgi Sert, Günther Greiner, and Jochen Süßmuth. Gpu based arap deformation using volumetric lattices. In *Eurographics (Short Papers)*, pages 85–88, 2012.
- [ZWA⁺21] Hongyu Zhou, Xing Wang, Wesley Au, Hanwen Kang, and Chao Chen. Intelligent robots for fruit harvesting: Recent developments and future challenges. 2021.

Appendix A

Comparison of the proposed methods

In this appendix, we aim to compare the proposed methodologies in this dissertation. To accomplish the mentioned objective, in the first step, the approaches described in chapters 3 and 4 are compared. To do that, two different simulations are conducted, and the results are presented in Fig. A.1 and Fig. A.2. The initial and final shapes of the two methods and the evolution of the states of the controlled point can be seen in those figures. The error of the two methods is obtained using (3.17).

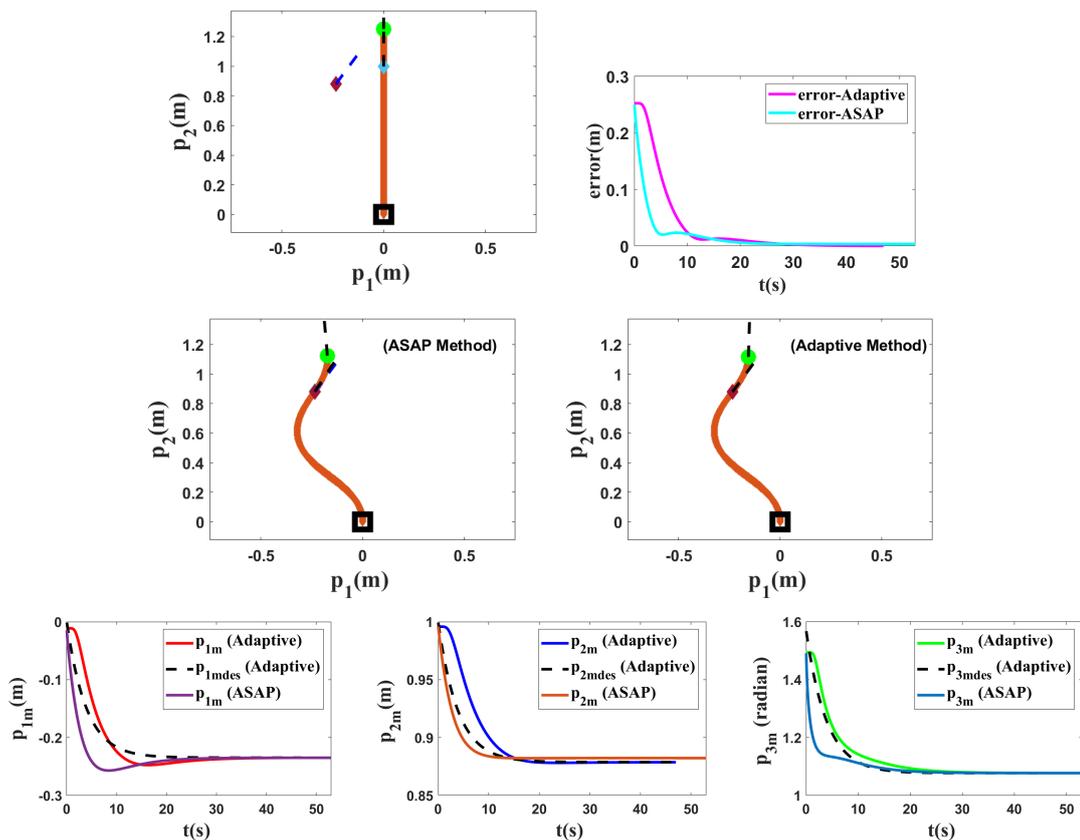


FIGURE A.1: Simulation 1. Comparison of the proposed methods in chapters 3 (ASAP) and 4 (adaptive). The length of the simulated object (L) is $1.25(m)$. From the top-left, the initial shape, the error of two methods, the final shapes of two methods, and the evolution of the states can be seen. The target point is shown in purple.

As one can see, both methods are capable of driving the controlled point toward the target in an acceptable time. KER is used to model the object in both of the simulations. We used the same number of iterations for both methods. However, the method presented in chapter 3 (ASAP) is faster. Its computational time is 88 FPS, while the computational time of the adaptive method is 76 FPS. However, as mentioned in the dissertation,

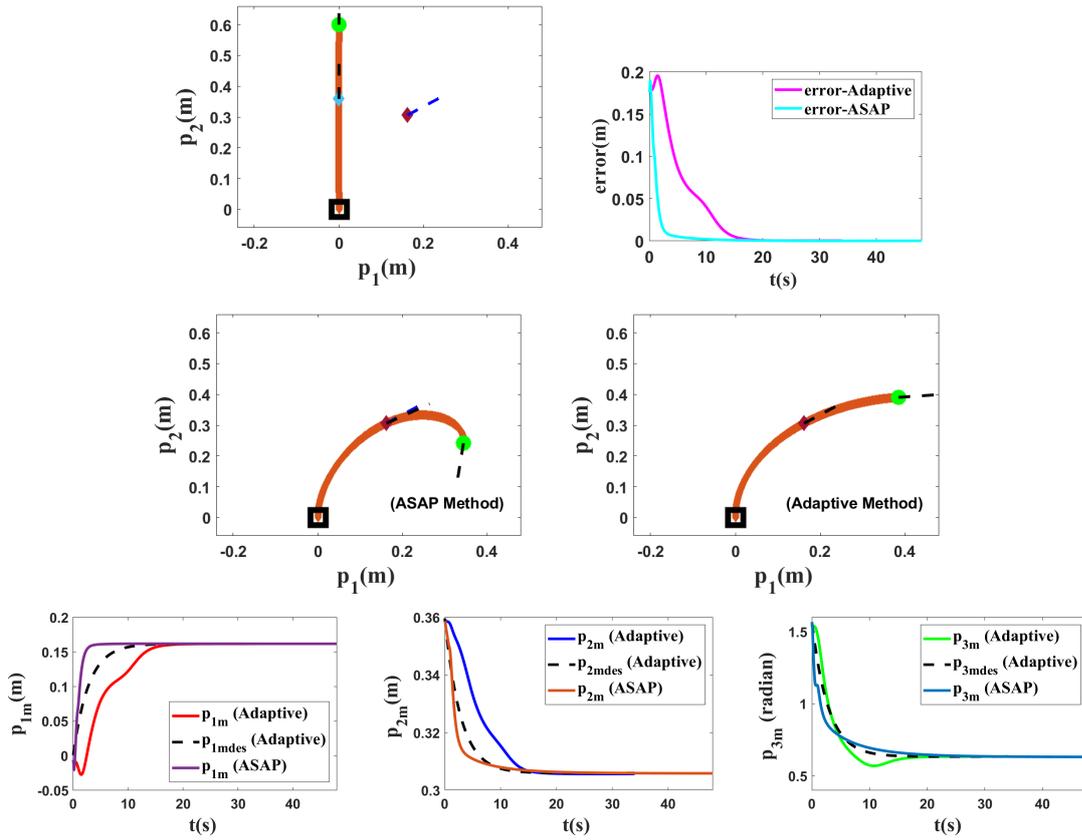


FIGURE A.2: Simulation 2. Comparison of the proposed methods in chapters 3 and 4. The length of the simulated object (L) is $0.60(m)$. From the top-left, the initial shape, the error of two methods, the final shapes of two methods, and the evolution of the states can be seen. The target point is shown in purple.

the main advantage of the adaptive method is that it can be used with a broader range of objects than the ASAP method. The initial and final state of the controlled points in two simulated tests and their final error are presented in Table A.1. Fig. A.3 shows some of the system parameters' estimations \hat{a}_{kj} (see chapter 4 equation (4.13)) in simulation 2. This figure presents how these parameters are changed during the object manipulation in this simulation using the adaptive method presented in chapter 4.

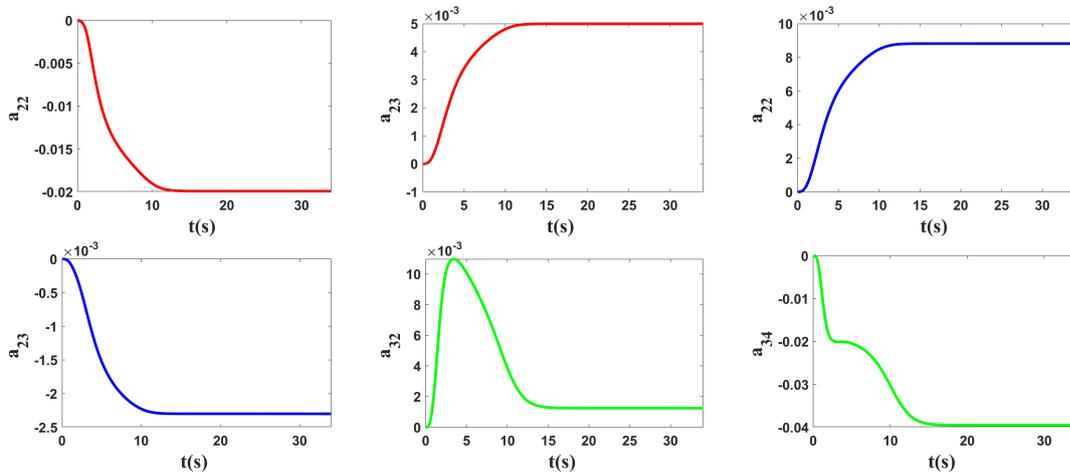


FIGURE A.3: Simulation 2. Some of the system parameters' estimations \hat{a}_{kj} .

TABLE A.1: State of the controlled points in the performed simulations.

Simulation number	Initial pose (m, m, rad)	Target pose (m, m, rad)	Final error (m):	
			ASAP	Adaptive
Sim. 1	[0.000, 1.000, 1.571]	[-0.235, 0.879, 1.077]	0.002	0.000
Sim. 2	[0.000, 0.360, 1.571]	[0.162, 0.306, 0.634]	0.000	0.000

Fig. A.4 to Fig. A.6 present the result of the comparison between chapters 5 and 6 in 2-D. ARAP is used to model the object in these simulations. The first row of the plots (i.e., Fig. A.4 to Fig. A.6) shows the initial and final shape of the object in these simulations. The second row presents the evolution of the error during these tests. The error of the two methods is obtained using (5.15). Even though both methods reach the target, the adaptive method is much faster. With the same number of iterations, the computational time of the adaptive method is 220 FPS, while the optimal method's computational time is 52 FPS. In the third row of the plots, we show how the adaptive method tracks its desired shape in a few sequential snapshots. The final error of the object's shape in the simulated tests is presented in Table A.2. To present how the system parameters' estimations \hat{a}_{kj} (see chapter 6 equation (6.15)) are changed, some of them are displayed in Fig. A.7 for simulation 5. This figure demonstrates these parameters' evolution during the object manipulation in this simulation using the adaptive method presented in chapter 6.

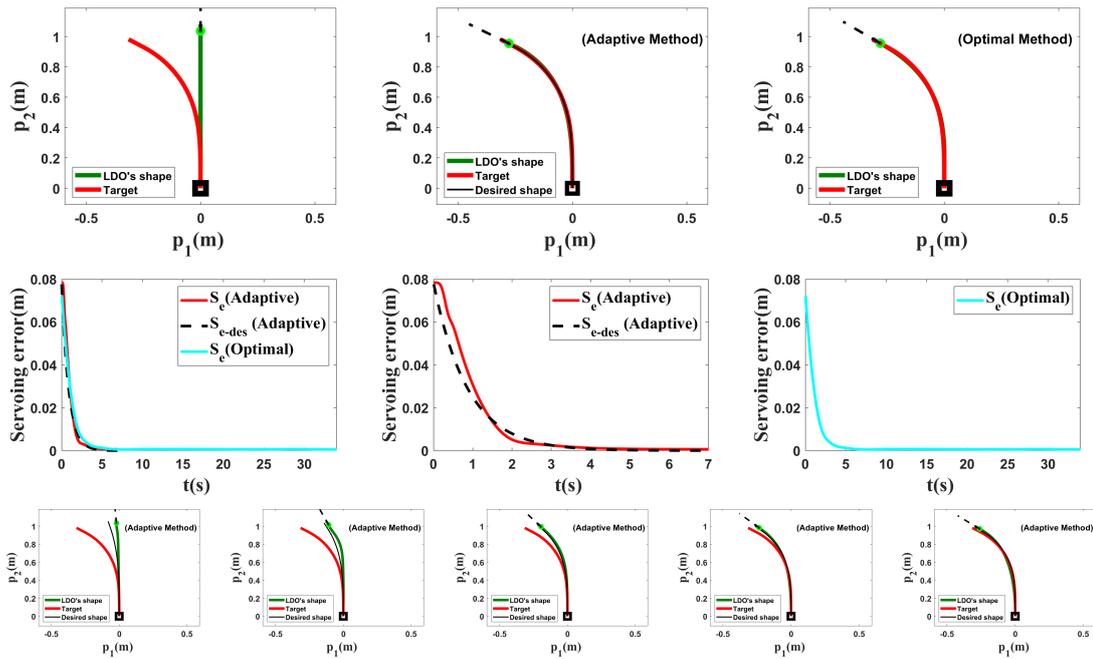


FIGURE A.4: Simulation 3. Comparison of the proposed methods in chapters 5 (optimal) and 6 (adaptive). $L = 1.08(m)$. The first row shows the initial and final shapes of the object. The second row demonstrates the evolution of the servoing error in these tests. In the third row, the object's deformation with respect to its desired shape is shown for the adaptive method.

We used the same number of iteration in each test and as one can see, the adaptive converged faster and has smaller computational cost than the optimal controller.

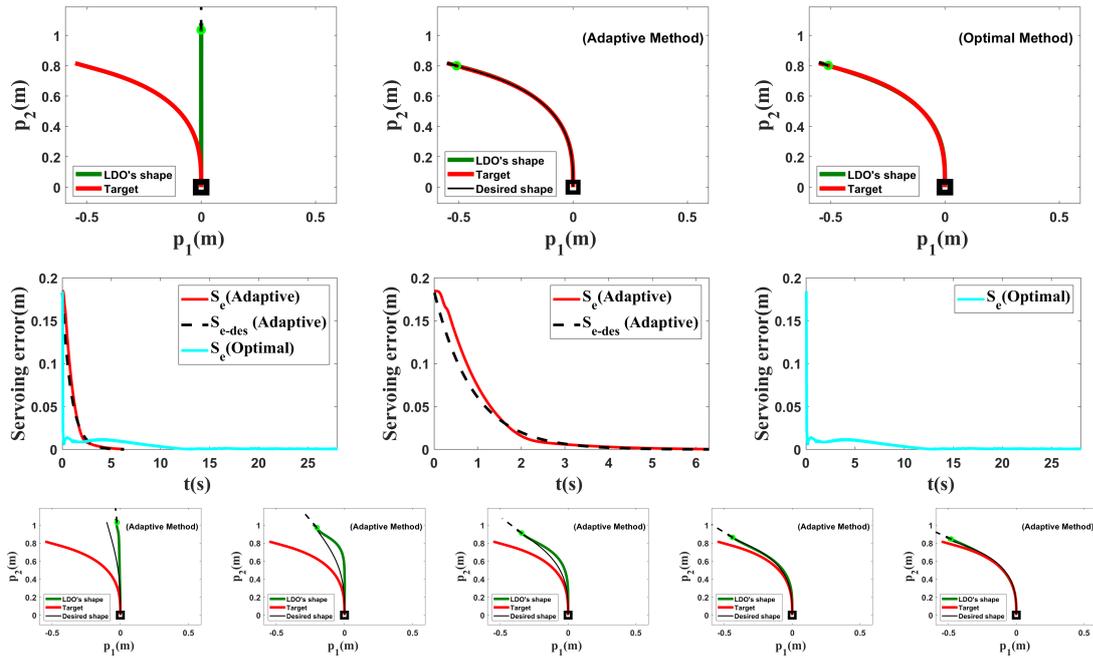


FIGURE A.5: Simulation 4. Comparison of the proposed methods in chapters 5 and 6. $L = 1.08(m)$. The first row shows the initial and final shapes of the object. The second row demonstrates the evolution of the servoing error in these tests. In the third row, the object's deformation with respect to its desired shape is shown for the adaptive method.

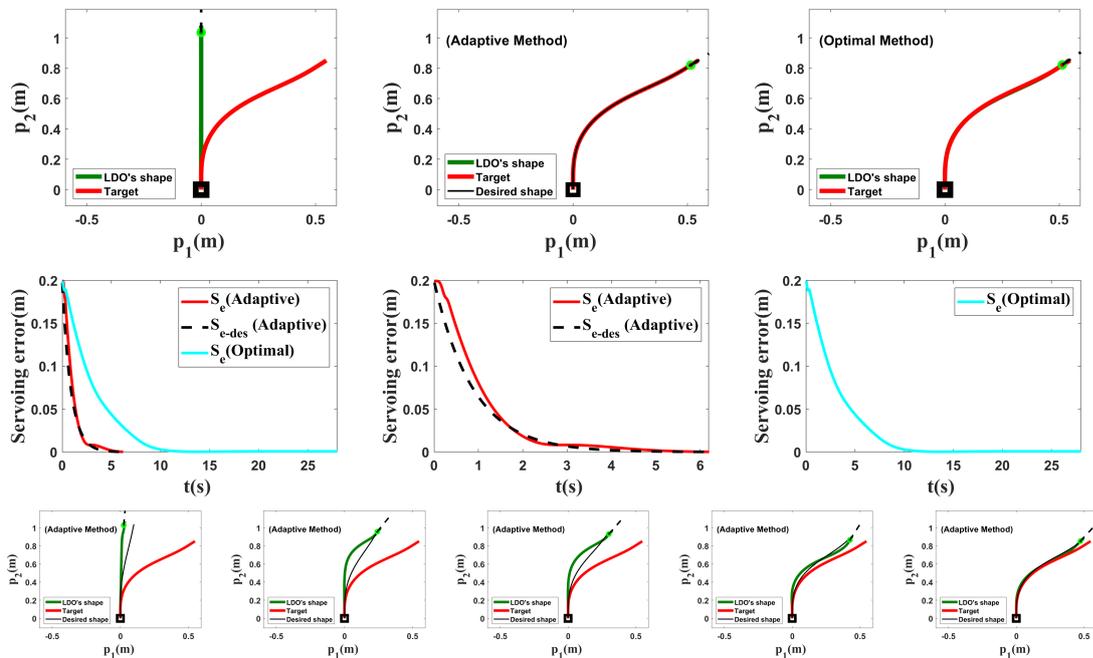
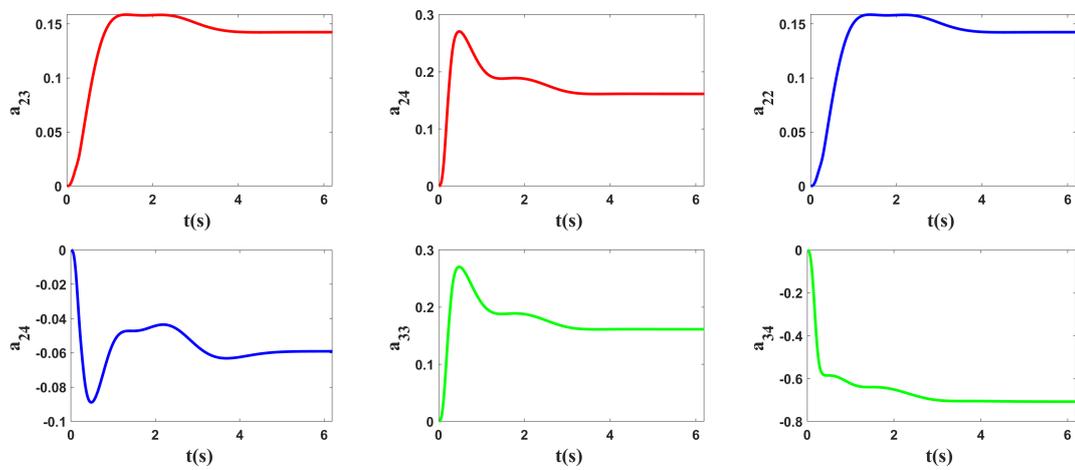


FIGURE A.6: Simulation 5. Comparison of the proposed methods in chapters 5 and 6. $L = 1.08(m)$. The first row shows the initial and final shapes of the object. The second row demonstrates the evolution of the servoing error in these tests. In the third row, the object's deformation with respect to its desired shape is shown for the adaptive method.

TABLE A.2: Final error of the performed tests.

Simulation number	Final servoing error (m):	
	Optimal	Adaptive
Sim. 3	0.001	0.001
Sim. 4	0.000	0.000
Sim. 5	0.001	0.000

FIGURE A.7: Simulation 5. Some of the system parameters' estimations \hat{a}_{kj} .