











Université Polytechnique des Hauts-de-France
LAMIH UMR CNRS 8201

École doctorale SPI
Université Lille Nord de France

Thèse préparée et soutenue par

Maxime Thieffry

le mercredi 16 octobre 2019 à Valenciennes
pour obtenir le grade de docteur en automatique

Model-Based Dynamic Control of Soft Robots

Commande dynamique de robots déformables basée sur un modèle numérique

Jury:

Véronique Perdereau, Professeure Sorbonne Université Paris	Présidente
Frédéric Boyer, Professeur IMT Atlantique, Nantes	Rapporteur
Nicolas Petit, Professeur Mines Paris Tech	Rapporteur
Christian Duriez, Directeur de recherche INRIA Lille	Examinateur
Alexandre Kruszewski, Maître de conférence Centrale Lille	co-Directeur
Thierry-Marie Guerra, Professeur UPHF Valenciennes	co-Directeur

Dimidium facti, qui coepit,
habet; sapere aude, incipe

Horace, Épitres I

Remerciements / Acknowledgements

First of all, I would like to thank everyone involved in this project from near and far. Thank you for your support without which I could never have reached this point.

I would also like to thank Prof. Frédéric Boyer and Prof. Nicolas Petit for reviewing this thesis and Prof. Véronique Perdereau for accepting to be part of my jury.

I am forever grateful to all the members of the Defrost team. I cannot judge the scientific interest of this thesis, but it has at least been the opportunity to meet some friends. I could never have done this work without your support, in the lab and outside. Thank you for everything that you have done for me during this last three years.

Special thanks to my – brilliant – advisors Alexandre Kruszewski, Christian Duriez and Thierry-Marie Guerra. I can not thank you enough for giving me the opportunity to complete this thesis.

Finally, I would like to thank Prof. Daniela Rus and Dr. Robert Katzschmann for welcoming me in their lab during summer 2018. It was a fantastic experience and a really great time.

Je tiens tout d'abord à remercier Frédéric Boyer et Nicolas Petit d'avoir accepté d'être rapporteurs de cette thèse ainsi que Véronique Perdereau d'avoir accepté de participer au jury.

Si une chose est sûre, c'est que cette thèse n'aurait jamais pu aboutir sans l'encadrement exceptionnel dont j'ai bénéficié durant ces trois années. Merci infiniment à Alex, Christian et Thierry-Marie. Merci de m'avoir donné ma chance et de me faire confiance depuis le début, merci pour votre support et votre encadrement au quotidien, merci de me faire sentir stupide et de me rappeler que beaucoup encore il me reste à apprendre.

Un immense merci à tous les membres de l'équipe Defrost: Anne, Alex, Bruno, Bruno, Christian, Damien, Fred, Jérémie, Gang, Mario, Mei, Olivier, Pierre, Stefan, Thomas, Thor, Ugo, Walid, Zhongkai, et tout ceux que j'oublie... Je n'aurais jamais pu terminer ce travail sans vous. Merci pour votre aide et votre soutien au quotidien, au travail et en dehors. Merci tout particulièrement à Anne sans qui j'aurais été perdu plus d'une fois.

Non, je ne vous ai pas oublié... De toute les décisions que Christian a prises ces trois dernières années, la meilleure a sans aucun doute été de m'installer dans le bureau 210. Eulalie et Félix, difficile de résumer trois ans en quelques lignes mais on peut objectivement se mettre d'accord sur deux points: vous avez partagé mon bureau (et pas l'inverse) et à chaque fois que vous avez eu l'impression de me battre, je vous ai laissé gagner. Ces trois dernières années auraient été beaucoup moins agréables sans vous. Merci pour tout.

Merci aux Thomas(s). Je ne sais même pas pourquoi je vous consacre un paragraphe, mais j'espère que vous apprécierez l'effort comme il se doit. On se retrouvera un jour à 3 sur un mont, au fond.

Merci également à Quentin Grandin et Taki-Eddine Korabi qui m'ont accompagné au début de mes études et sans qui je n'aurais peut-être jamais commencé ce travail.

Finalement, pensées spéciales pour toute ma famille. Vous êtes trop nombreux pour êtes tous cités mais je ne vous remercierai jamais assez. Maman et Papa, merci infinement, pas spécialement pour votre aide pour ce travail mais pour tout le reste, ce qui est sans doute encore plus important. Marine, Baptiste et Basile, merci pour tout.

Évidemment, les derniers mots seront pour toi Georgie. Merci pour tout final.

Abstract

This thesis focuses on the design of closed-loop control laws for the specific needs of dynamic control of soft robots, without being too restrictive regarding the robots geometry. It covers the entire development of the controller, from the modeling step to the practical experimental validation. In addition to the theoretical studies, different experimental setups are used to illustrate the results. A cable-driven soft robot and a pressurized soft arm are used to test the control algorithms. Through these different setups, we show that the method can handle different types of actuation, different geometries and mechanical properties. This emphasizes one of the interests of the method, its genericity.

From a theoretical point a view, large-scale dynamical systems along with model reduction algorithms are studied. Indeed, modeling soft structures implies solving equations coming from continuum mechanics using the Finite Element Method (FEM). This provides an accurate model of the robots but it requires to discretize the structure into a mesh composed of thousands of elements, yielding to large-scale dynamical systems.

This leads to work with models of large dimensions, that are not suitable to design control algorithms. A first part is dedicated to the study of the large-scale dynamic model and its control, without using model reduction. We present a way to control the large-scale system using the knowledge of an open-loop Lyapunov function. Then, this work investigates model reduction algorithms to design low order controllers and observers to drive soft robots.

The validated control laws are based on linear models. This is a known limitation of this work as it constrains the guaranteed domain of the controller. This manuscript ends with a discussion that offers a way to extend the results towards nonlinear models. The idea is to linearize the large-scale nonlinear model around several operating points and interpolate between these points to cover a wider workspace.

Keywords: Soft Robotics, Robust Control, Linear Matrix Inequality, Finite Element Method, Large-scale Models, Model Reduction

Résumé

Cette thèse s'intéresse à la modélisation et à la commande de robots déformables, c'est à dire de robots dont le mouvement se fait par déformation. Nous nous intéressons à la conception de lois de contrôle en boucle fermée répondant aux besoins spécifiques du contrôle dynamique de robots déformables, sans restrictions fortes sur leur géométrie. La résolution de ce défi soulève des questions théoriques qui nous amènent au deuxième objectif de cette thèse: développer de nouvelles stratégies pour étudier les systèmes de grandes dimensions.

Ce manuscrit couvre l'ensemble du développement des lois de commandes, de l'étape de modélisation à la validation expérimentale. Outre les études théoriques, différentes plateformes expérimentales sont utilisées pour valider les résultats. Des robots déformables actionnés par câble et par pression sont utilisés pour tester les algorithmes de contrôle. À travers ces différentes plateformes, nous montrons que la méthode peut gérer différents types d'actionnement, différentes géométries et propriétés mécaniques. Cela souligne l'un des intérêts de la méthode, sa généricité.

D'un point de vue théorique, les systèmes dynamiques à grande dimensions ainsi que les algorithmes de réduction de modèle sont étudiés. En effet, modéliser des structures déformables implique de résoudre des équations issues de la mécanique des milieux continus, qui sont résolues à l'aide de la méthode des éléments finis (FEM). Ceci fournit un modèle précis des robots mais nécessite de discrétiser la structure en un maillage composé de milliers d'éléments, donnant lieu à des systèmes dynamiques de grandes dimensions.

Cela conduit à travailler avec des modèles de grandes dimensions, qui ne conviennent pas à la conception d'algorithmes de contrôle. Une première partie est consacrée à l'étude du modèle dynamique à grande dimension et de son contrôle, sans recourir à la réduction de modèle. Nous présentons un moyen de contrôler le système à grande dimension en utilisant la connaissance d'une fonction de Lyapunov en boucle ouverte. Ensuite, nous présentons des algorithmes de réduction de modèle afin de concevoir des contrôleurs de dimension réduite et des observateurs capables de piloter ces robots deformables.

Les lois de contrôle validées sont basées sur des modèles linéaires, il s'agit d'une limitation connue de ce travail car elle contraint l'espace de travail du robot. Ce manuscrit se termine par une discussion qui offre un moyen d'étendre les résultats aux modèles non linéaires. L'idée est de linéariser le modèle non linéaire à grande échelle autour de plusieurs points de fonctionnement et d'interpoler ces points pour couvrir un espace de travail plus large.

Mots-clés: Robots Déformables, Commande Robuste, Inégalité Matricelle Linéaire, Méthode des Éléments Finis, Modèles de Grande Dimension, Réduction de Modèle

Contents

Remerciements / Acknowledgements iii

Abstract v

Résumé vii

Contents ix

Notations and Acronyms xiii

I Introduction 1

1 General Introduction 3
- 1.1 From rigid to soft robots . . . 3
- 1.2 Link between Soft Robots and Large-Scale Dynamical Systems . 8
- 1.3 Present Contributions . . . 9

2 State of the Art: Dynamic Control of Soft Robots 11
- 2.1 Soft Robot Design, Actuation & Sensing . . . 12
- 2.2 Modeling . . . 12
- 2.3 Control . . . 17
- 2.4 Some recent works about control theory . . . 23

II Large-Scale Model and Controller 25

3 Finite Element Model 29
- 3.1 Nonlinear second order model . . . 30
- 3.2 Nonlinear state-space equation . . . 30
- 3.3 Linear large-scale state-space equation . . . 32
- 3.4 Illustrations . . . 35

4 Large Scale Feedback Controller **43**
4.1 Theoretical basis . . . 44
4.2 Energy-based Lyapunov function . . . 45
4.3 Parameterized Energy Function . . . 46
4.4 Simulation results . . . 48

III Model Order Reduction and Low-order Observer **51**

5 Reduced Order Models **55**
5.1 Presentation of Reduction Algorithms . . . 56
5.2 Low Order Model . . . 61
5.3 Comparison of Reduction Methods . . . 62

6 Low-order Observer **67**
6.1 Introduction . . . 67
6.2 Discrete-time Unkown Input Observer Design . . . 68
6.3 Observer design according to modeling error issue . . . 69
6.4 Summary on Observer Design . . . 70

IV Low-dimensions Model-based Controller **71**

7 Low-dimension Controller with Large-Scale Stability using Energy-based Lyapunov Functions **77**
7.1 Closed-loop large-scale Lyapunov function . . . 78
7.2 Simulation Experiments . . . 79
7.3 Real-time Experiments . . . 82

8 Trajectory Tracking for Large-Scale Linear Systems **85**
8.1 Introduction . . . 86
8.2 Control Design . . . 87
8.3 Computation of Controller Gains . . . 90
8.4 Experimental Validation . . . 94

V Perspectives and Conclusions **99**

9 Increased Guaranteed Domain Of Dynamic Controller: Towards Nonlinear Models **103**
9.1 Introduction: basic concepts of LPV systems . . . 104
9.2 Collection of Linear Systems . . . 105

9.3 Linear Parameter Varying (LPV) Models 106
9.4 Discussion and Future Work 109

10 Dynamic Control with Contacts with the Environment 111
10.1 Dynamic Model with contacts 112
10.2 Controller Design using Switched Systems Theory 113
10.3 Inverse Simulation-based Controller 114
10.4 Summary and Future Work 115

11 General Conclusion 117

A Preliminary in Systems Theory and Control 121
A.1 Introduction 121
A.2 Properties of LTI Systems 122
A.3 Technical Results in Linear Algebra 125

Bibliography 127

Notations and Acronyms

$\mathbb{N}$	Set of integers
$\mathbb{R}$, $\mathbb{R}_+$	Set of real numbers, nonnegative real numbers
$\mathbb{R}^{n\times p}$	Matrix algebra of dimensions $n \times p$ with coefficients in $\mathbb{R}$
$\mathbf{M}$, $\mathbf{K}$, $\mathbf{D}$	Mass, Stiffness and Damping matrices coming from FEM model
I_n	Identity matrix of dimension $n \times n$
$\dot{x}$	Derivative of x with respect to time
x_+	Value of x at next time step
$\Delta V(x)$	Stands for $V(x_+) - V(x)$
A^T	Transpose of matrix A
A^{-1}	Inverse of matrix A
$A > 0$	Symmetric positive definite matrix
$\|x\|$	Euclidian norm of vector x
$(*)$	Transpose quantity in an expression or in a symmetric matrix
$A^TP + (*)$	Stands for $A^TP + PA$
$\begin{pmatrix} P & * \\ A & P \end{pmatrix}$	Stands for $\begin{pmatrix} P & A^T \\ A & P \end{pmatrix}$

BT	Balanced Truncation
FEM	Finite Element Method
FOM	Full Order Model
IRKA	Iterative Rational Krylov Algorithm
LMI	Linear Matrix Inequality
LPV	Linear Parameter Varying
LTI	Linear Time Invariant
MIMO	Multi Input Multi Ouput
MOR	Model Order Reduction
POD	Proper Orthogonal Decomposition
ROM	Reduced Order Model
SISO	Single Input Single Output
SVD	Singular Value Decomposition

Part I

Introduction

Chapter 1

General Introduction

The purpose of this manuscript is double, the first objective is to provide new methodologies to model and control soft robots. Solving this first challenge raises theoretical questions that lead us to the second objective of this thesis, developing new strategies to study large-scale systems. Let us begin with a bit of history and context about soft robotics and then explain the link between soft robots and large-scale dynamical systems.

Contents

1.1 From rigid to soft robots 3
1.2 Link between Soft Robots and Large-Scale Dynamical Systems . 8
1.3 Present Contributions 9

1.1 From rigid to soft robots

Different definitions of robotics can be found in the litterature but a general one is given by Jean-Paul Laumond (Laumond 2012):

> Robotics concerns the relationship that a machine which moves, and whose motions are controlled by a computer, can have with the real world. In this sense the robot differs from automats, whose motions are mechanically determined, and computers, which manipulate information but do not move.

The concept of robotics is also inherently related to the notions of motion and control. How does a human, a user, control the motion of a robot? This very basic question is somehow the first idea of this thesis. One could trace back the history of robotics further but the first industrial robot, the Unimate of General Motors, was used in 1961. Since then, robotics has been widely used in the manufacturing industry, where it plays a large part in the organization of the means of production. The repetitive nature of the tasks a robot is asked to perform in a well-structured environment is a key-point of its success. Robots are no longer restricted to manufacturing industry but are widely spread in the every-day life, which increases the interest for robotics. Nowadays, even if there are still challenges and open questions about traditional rigid robots, the underlying mechanics, the modeling and the simulation are well understood and many results are available concerning these topics.

The ever-increasing mastery of these robots, the desire to design robots to interact with humans and/or in confined space in contact with the environment, pushed the robotics community to invent new paradigms, among which is soft robotics. Biology provides a major inspiration in the design of soft robots. The elephant trunk, octopus arm or snake body lead to the design of the first soft robots (Majidi 2014, Kim et al. 2013).

By soft robots, we mean robots made of deformable materials whose motion are obtained by deformation and where no joints are presents in the structure. Soft robots are a sub-class of continuum robots, that are hyper-redundant robots. If one keeps repeating the process of adding joints until their number approaches infinity, the robot will converge to what is called a continuum robot (Morales Bieze 2017).

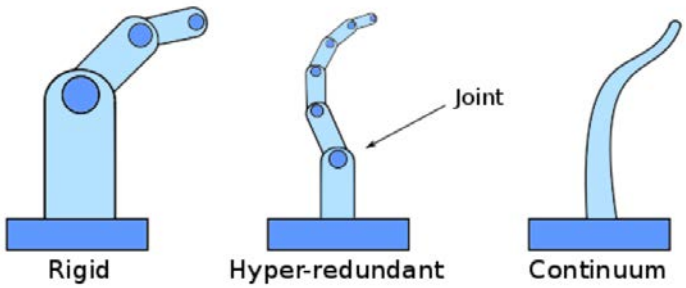


Figure 1.1: *From rigid to continuum robots, from (Morales Bieze 2017).*
Left: Rigid robot, Center: Hyper-redundant robot, Right: Continuum robot.

Where there is no need for a high level of adaptability in the environment, there is no need to think about soft robots. One of the driving forces behind soft robotics is the need for machines that can work closely with humans rather than

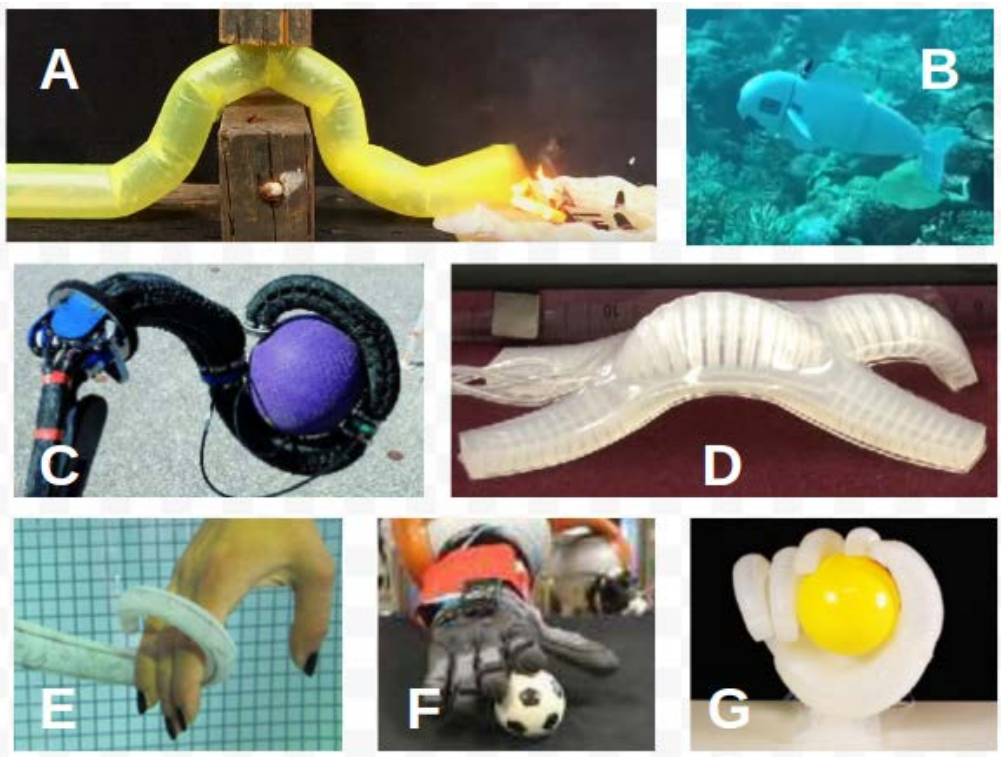


Figure 1.2: *(Non-exhaustive) examples of soft robotics systems.*
A) growing soft robots (Greer et al. 2019), B) soft robotic fish (Katzschmann et al. 2016), C) continuum manipulator (McMahan et al. 2006), D) multi-gait soft robot (Shepherd et al. 2011), E) soft arm inspired by the octopus (Calisti et al. 2010), F) Anthropomorphic soft hand (Della Santina et al. 2019), G) Underactuated robotic hand (Deimel & Brock 2016).

in competition with them. A traditional rigid robot will most likely outperform a soft robot if the desired task is well defined, in a well known and structured environment.

By definition, soft robots are compliant to the environment, and regarding the variety of materials used for their design and manufacture, they are also lighter than rigid robots. Taking advantage of the light weight of the structure, soft robots could exhibit a bigger power to weight ratio, which can make them easy to deploy. Soft robots also promise disruptive advances in many field, such as identified in (Lamnabhi-Lagarrigue et al. 2017) as one *"examples of high-impact Systems & Control applications in the coming decades"*.

Potential applications are too numerous to be exhaustive as they cover all the fields of applications of traditional robotics. Special attention may be given to

healthcare industry, in particular surgery (Cianchetti et al. 2014, Slade et al. 2017), prosthetics and artificial organs, agricultural work (Scimeca et al. 2019, Hughes et al. 2018), search and rescue (Tolley et al. 2014), inspection in confined areas (Chablat et al. 2019), or simply home assistants... possibilities to use soft robots are boundless. The few following examples show the interdisciplinarity of soft robotics as well as possible applications.

Surgery

The application that will perhaps highlight the most soft robots is surgical robotics. Thinking about robotic surgery tasks, contacts between human tissues and the robots are required. It is likely that it will be less dangerous if the tissues are in contact with a soft robot rather than a rigid one. Robotic surgery has been studied and used for a while, as recalled in (Taylor et al. 2016, Ballantyne 2002, Lanfranco et al. 2004), but soft robotic surgical is most recent, one of the first use is in (King et al. 2008). Minimal invasive surgery is a hot topic, however most of the instruments used are rigid and lack a sufficient number of degrees of freedom. Due to their compliant behavior, soft robots are a promising technology to overcome these issues. A stiffness-controllable STIFF-FLOP soft arm is presented in (Cianchetti et al. 2014) to overcome the current limitations in surgical instrumentations. Authors of (Abidi et al. 2018) present a dexterous 2-module soft robot used as a laparoscopic tool. An inverse problem, solved based on a constant curvature model, allows the surgeon to control the robot and navigate safely around the organs. In addition, a soft catheter capable of apical extension is presented in (Slade et al. 2017), it is pre-shaped to patient specific trajectories and can reach constrained location of the human body while applying low forces on the tissues.



Figure 1.3: *Growing robots used for catheter navigation in (Slade et al. 2017).*

Assistive medical devices, soft wearable devices

Recently, a soft robotic implant has been designed for cardiac assistance in (Payne et al. 2017). In this work, authors showed that such a device could be a credible alternative for heart failure therapies. McKibben-based actuators and pressure catheter sensors are used to control the device. This work also raises future challenges to be solved, among which are the need to develop soft robotic sensing and control technologies, as well as the miniaturization of the control system.

A second example of soft medical device is the development of soft exoskeletons like the soft exosuits developped at Harvard Biodesign Lab (Ding et al. 2017, O'Neill et al. 2017). An exoskeleton for rehabilitation is presented in (Vigne et al. 2018), where the authors use a sensor-based method to estimate the flexibilities of an articulated system. Removing rigid links in wearable robots would make them more comfortable to wear, and therefore would make their use more efficient. Figure 1.4 presents the soft robotic glove designed in (Polygerinos et al. 2015).



Figure 1.4: *Soft robotic glove presented in (Polygerinos et al. 2015).*

Fragile and/or unstructured manufacturing

Automation of manufacturing facilities has for now been limited to structured tasks, leading many industries to rely on hand assembly when the environment is not structured enough, i.e. when objects to manipulate are with different shapes, weights and sizes. Moreover, traditional grippers may not be able to manipulate fragile objects without damaging them. There is a need to develop soft grippers that could deal with, for instance, food items or perfume bottles (Brown et al. 2010, Hao et al. 2016). Figure 1.5 shows a universal soft gripper detailed (Brown et al. 2010) designed to grasp any type of items.



Figure 1.5: *Universal soft gripper presented in (Brown et al. 2010).*

1.2 Link between Soft Robots and Large-Scale Dynamical Systems

From biology or fluid mechanics, applications involving a large number of variables are widely spread in the nature. In addition to these natural examples, engineers use numerical simulation to design engineering applications. Indeed, it shortens the development time of engineering systems as different designs, actuation or sensing methods and control approaches can be tested out quickly. For computer-based simulations, the demand for a highly accurate description of realistic phenomena leads to work with systems of very large dimension.

When the systems studied are part of a control loop, one is interested in the reaction of this system with respect to a given input. The control of large-scale dynamical systems has been widely studied in the literature. To study the stability of such systems, large-scale Lyapunov and Riccati equations have been studied in (Benner & Saak 2013, Haber & Verhaegen 2016).

Instead of directly studying the large-scale models, model reduction algorithms permit to study a low order approximation of the models. Model reduction is also an active field of research and many results are available as in (Benner et al. 2017, Astolfi 2010).

Soft robotics is not directly linked to large-scale systems. However, computing a model of soft robots is a challenging task, it requires to solve equations from continuum mechanics involving nonlinear materials. To address this task, numerical methods such as the finite element method can come at hand. This method requires a spatial discretization of the structure into a mesh. For the model to be precise, it requires a large number of variables and thus yields to work with large-scale systems.

In addition to constraints related to modeling of soft structures in real time, the large dimension of the system studied makes it difficult to design a controller.

Using classical tools of automatic control, such as H_∞ attenuation, LQR-LFT, the design of a low dimension (few inputs, outputs) linear system, including performances, has nowadays reached a high level of maturity. However, when the number of decision variables increase, so does the computational cost and the control design will imply further developments that can highly complicate the satisfaction of the required constraints.

1.3 Present Contributions

To make full use of the aforementioned advances, some challenges still need to be solved. A detailed list of them can be found in chapter 2, which gives a review of recent results dealing with the design, model, simulation and control of soft robots.

This thesis focuses on the design of closed-loop control laws for the specific needs of dynamic control of soft robots, without assumptions about the robots geometry. To achieve this objective, a dynamic model of the soft structure is presented in chapter 3. It relies on Finite Element Method (FEM) to solve continuum mechanics equations numerically (Coevoet et al. 2017). This provides an accurate model of the robots but it requires to discretize the structure into a mesh composed of thousands of elements, yielding to large-scale dynamical systems. Chapter 4 presents a way to control the large-scale system using the knowledge of an open-loop Lyapunov function (Thieffry et al. 2018). Then, chapter 5 details the model reduction algorithms used to reduce the dimensions of the system and chapter 7 deals with the control of the low order model obtained thanks to this reduction step (Thieffry et al. 2019, Katzschmann, Thieffry et al. 2019). A trajectory tracking control design is presented in chapter 8, where both feedforward and feedback elements are used in the controller[1].

In addition, chapter 9 offers a way to get rid of the linearity assumption using linear parameter varying (LPV) systems. Finally, chapter 10 presents tracks to extend the results to handle contacts between the robots and their environment. Chapter 11 recalls the main contributions of this thesis to conclude this manuscript.

[1]These results have been accepted for publication: *Trajectory Tracking Control Design for Large Scale Linear Dynamical Systems with applications to Soft Robotics, M. Thieffry, A. Kruszewski, T.M. Guerra, C. Duriez, in IEEE Transactions on Control Systems Technology*

Chapter 2

State of the Art: Dynamic Control of Soft Robots

In their article published in 2008, authors of (Trivedi et al. 2008) identified future research challenges to pursue the development of soft robotics. Among them lies the development of soft sensors and actuators as well as soft robots design, modeling and control techniques. Since this publication, the interest of the robotics community for soft robots has grown but most of the previously cited challenges are still fully open. In this chapter some contributions on these topics are gathered.

Contents

2.1 Soft Robot Design, Actuation & Sensing 12
2.2 Modeling 12
2.2.1 Piece-wise Constant Curvature 13
2.2.2 Cosserat Theory 14
2.2.3 Finite Elements Method 15
2.2.4 Reduced order modeling of soft robots 16
2.2.5 Summary on modeling approaches 17
2.3 Control 17
2.3.1 Open-loop 18
2.3.2 Closed-loop 20
2.4 Some recent works about control theory 23

2.1 Soft Robot Design, Actuation & Sensing

To take full benefits of the robots compliance, actuation and sensing methods should not bring additional stiffness nor constrain the robots motion. Soft actuation and sensing is also a full research topic inside the domain of soft robotics.

Soft robot designs are often inspired by natural organisms such as snakes, worms or elephants trunk (Calisti et al. 2011, Kim et al. 2013, Katzschmann et al. 2016). Deformable robots are also studied to create self-folding structures (Felton et al. 2014). This method inspired from origami creates complex shapes scalable to different sizes (Onal et al. 2013).

Different types of actuators are used to drive soft robots such as cables, pneumatics or hydraulics actuators, electroactive polymers etc. Figure 2.1 shows an example of actuation type for different soft robots. In addition, one of the objectives is to integrate the actuator in the soft structure which can also be printed with the soft body such as in (Fries et al. 2014) or (Niiyama et al. 2015). In the meantime, recent works show how to integrate soft sensors into the soft structure (Felt et al. 2017, Truby et al. 2019). Figure 2.2 shows a sample of sensor methods available for soft robotics applications. In this manuscript, two different robots are used: a cable-driven and a pressurized soft robot; their design is detailed in chapter 3.4.

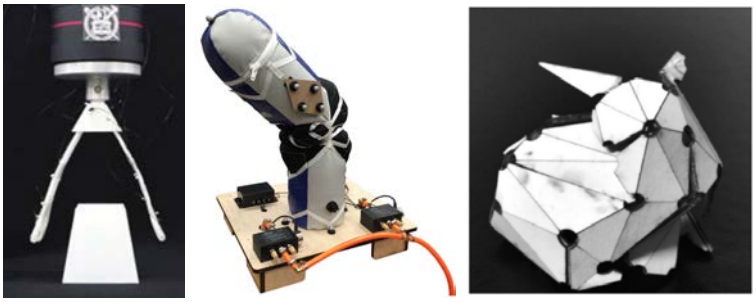


Figure 2.1: *Examples of actuation for soft robots.*
Left: Shape memory allow-based soft gripper from (Wang & Ahn 2017), Center: Inflatable soft arm from (Gillespie et al. 2016); Self-folding origami from (An et al. 2018).

2.2 Modeling

Soft robots have an theoretical infinite number of degrees of freedom. Developing a mathematical model suitable to describe the dynamics of such infinite degrees of freedom robots is still a challenging task. A model of soft robots should

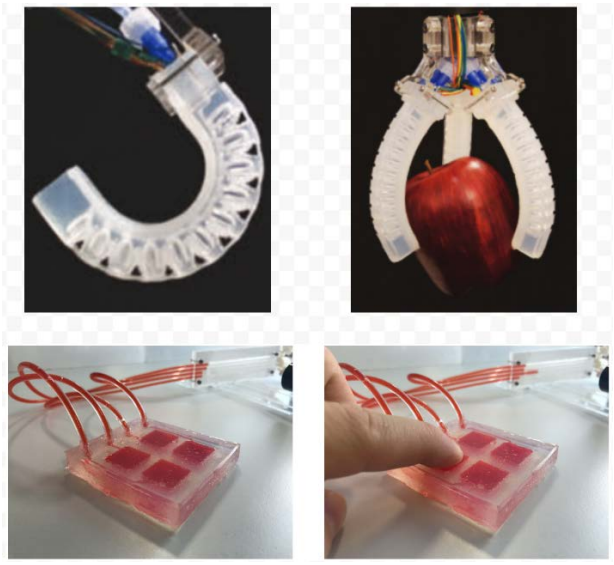


Figure 2.2: *Top: Soft fingers with embedded ionogel sensors (Truby et al. 2019); Bottom: soft robotic skin based on fluidic transmission (Soter et al. 2019).*

be at the same time, computationally affordable and sufficiently accurate. Different methods have been proposed to solve this challenge. From shape-specific methods to more generic ones, this section provides an overview of these methods.

2.2.1 Piece-wise Constant Curvature

Constant curvature modeling approach is the most used in the soft robotics community, as most of the robots are composed of beam elements, for which making the assumption of constant curvature is a possibility. For a complete review on constant curvature modeling, please refer to (Webster & Jones 2010)

This method eases the modeling of soft structure due to the assumptions in the model. It represents the structure with a fixed number of arcs described by three parameters: the radius of curvature, the angle of the arc and the bending plane.

It has been successfully applied to many continuum robots with different actuation systems such as cables or pneumatic chambers (Marchese et al. 2014). The Piece-wise constant curvature method decomposes the robot into a fixed number of continuously deformable segments with curvature constant in space but variable in time. Initially developed for kinematics studies, this method has been extended to dynamics modeling and used in closed-loop control experiments in (Katzschmann, Della Santina, Toshimitsu, Bicchi & Rus 2019).

Consider a robot composed by n segments, with a frame S_i attached at the end of each segment. The robot kinematics is described by n transformations $T_0^1, \ldots, T_{n-1}^n$ that link each frame to the following one. Figure 2.3 shows an example for a robot made of 3 sections. The resulting dynamic writes:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + D(q)\dot{q} + G(q) + Kq = A(q)\tau \tag{2.1}$$

where M is the inertia matrix, C is the centrifugal and Coriolis matrix, G is the gravitational field, D is the damping matrix, K is the stiffness matrix, A is the actuation matrix and τ is the joint torques. See (Della Santina et al. 2018, Katzschmann, Della Santina, Toshimitsu, Bicchi & Rus 2019) for a detailed formulation.

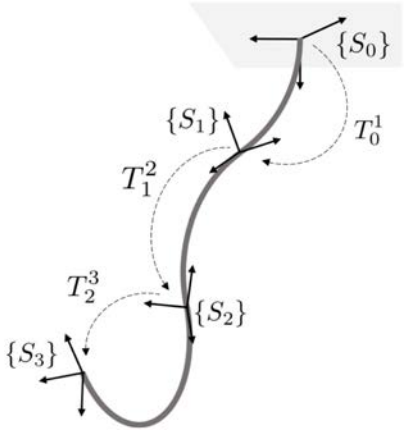


Figure 2.3: *Example of 3D piece-wise constant curvature robot composed by* 3 *sections from (Katzschmann, Della Santina, Toshimitsu, Bicchi & Rus 2019).*

2.2.2 Cosserat Theory

Cosserat theory provides a geometrically exact method to model soft robots, taking into account large deformations and displacements and handling material nonlinearities (Trivedi, Lotfi & Rahn 2008). As for the PCC method, it is intended for beam-like robots.

The Cosserat approach is an infinite degrees of freedom model where the structure is made of an infinite number of infinitesimal microsolids. To use this method in practice, the same idea as for the PCC method is applied to discretize the continuous model built with the Cosserat theory, yielding to a model completely described by a finite set of strain vectors. (Renda et al. 2017).

The configuration of a soft robot is given by the position vector q and an orientation matrix R, parameterized by the material abscissa $X \in [0, L]$ along the robot arm. The configuration space is defined as a curve $g(X)$ with:

$$g(X) = \begin{pmatrix} R(X) & q(X) \\ 0 & 1 \end{pmatrix} \tag{2.2}$$

The strain state is defined by the vector field along the curve $g(X)$ given by $\xi(X) = g^{-1}\partial g/\partial X = g^{-1}g'$, where $'$ represents the derivative with respect to space. The dynamic equation is similar to equation (2.1) with $\xi = q$. Figure 2.4 shows a schematic view of a robot modeled with Cosserat theory.

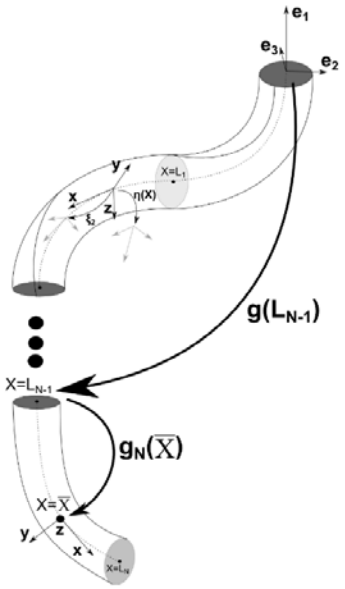


Figure 2.4: *Schematic of the kinematics of the piece-wise constant strain model from (Renda et al. 2016).*

2.2.3 Finite Elements Method

The Finite Element Method (FEM) consists in discretizing the structure into a finite number of small elements. The underlying equations coming from continuum mechanics are then solved for each of these elements. A modeling and simulation software dedicated to soft robots is implemented upon the open-source framework SOFA (Allard et al. 2007) along with a plugin for the specific needs of soft robots[1]. This modeling method can handle most of the geometries, provided that a CAD file for the structure exists.

[1]Soft Robots Plugin, https://project.inria.fr/softrobot/, see (Coevoet et al. 2017)

The theoretical foundations of this modeling framework are the ones of continuum mechanics for the material modeling, Lagrangian multipliers for constraints solving and Signorini's law for contacts (Coevoet et al. 2017).

To compute the internal forces, one needs to pick a deformation law. The relationship between the loads and resulting deformations is the constitutive equation, the Hooke's law is a common equation that makes the assumption of linearity of material response to strain. Other laws exist to express nonlinear strain- stress relationship, plastic deformations, brittle or hysteresis behavior. Different laws have different computational costs and one has to carefully choose the law that fits best the needs and computation time constraints.

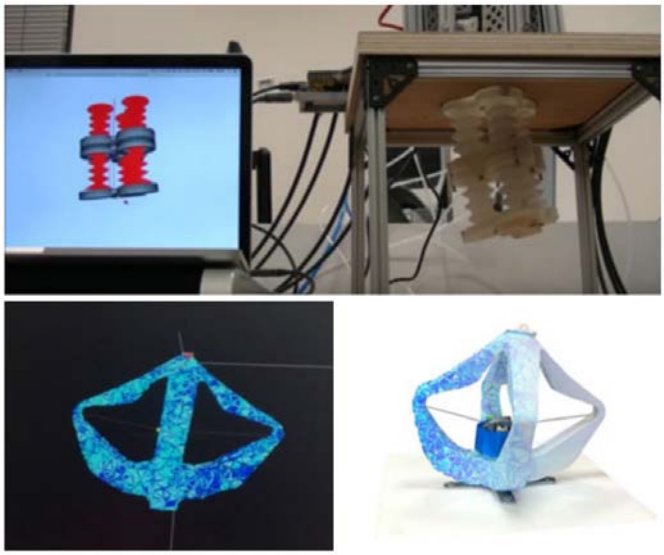


Figure 2.5: *Examples of soft robots modeled and control via a FEM model, from (Coevoet et al. 2017). Both pictures show the simulated model (left) along with the real robots (right).Top: pressurized soft robot. Bottom: cable-driven soft robot.*

2.2.4 Reduced order modeling of soft robots

Simulation offers many advantages for robotic applications. However, as said above, soft robots have an infinite number of degrees of freedom, that makes their modeling challenging. To tackle this issue, model order reduction is widely used in the computational mechanics and control communities, even if the objectives may be different.

On the one hand, one aims at reducing the computation time of the simulation process to reach real-time performances. On the other hand, one may want to preserve a particular input-output relationship to design a specific controller.

For both applications, the general objective is to find a model with few variables that describes the full order model in the sense of a given measure but different objectives may lead to different low-order models.

For soft robotics applications, recent work has been done to build a real-time model of soft robots compatible with control requirements (Chenevier et al. 2018, Goury & Duriez 2018). In addition, recent work presented in (Sadati et al. 2019) presents a comparison between two models for continuum manipulators: a reduced order model approach and a discretized model based on Euler-Bernouilli beam segments and relative states. Their new modeling approach presents many advantages: accuracy, real-time performances and enables a simple control design.

2.2.5 Summary on modeling approaches

Different strategies exist to model soft robots, each comes with its advantages and drawbacks. PCC and Cosserat theory offer an efficient model that is easy to implement and based on which controllers have successfully been designed. However, not every geometry can be modeled with these assumptions. Moreover, for PCC the models do not account for the material properties of the robot which plays a role in the kinematics of soft robots, in particular with heterogeneous materials.

With FEM, it is possible to define a precise model that represents faithfully the soft structure. In addition, the SOFA framework provides an efficient simulation engine to model these robots. However, this modeling technique is computationally expensive as it requires a precise mesh of the structure. To tackle this issue, GPU-based algorithms or model order reduction methods may be used. In addition to the assumption of linearity of material response to strain, in the following of this manuscript we limit the deformation cases to purely elastic behavior: the robot goes back to its initial shape when the actuation is released and the parameters of the materials are given by the Young's modulus and the Poisson's ratio of the Hooke's law.

2.3 Control

This section recalls existing methods dealing with soft robots control, it shows the large variety of approaches: open or closed-loop methods, learning algorithms or model-based controller etc. These different methods are complementary to each others: a good controller consists of a good feedforward action (open-loop) and a good feedback part (closed-loop). To tune these two parts, different methods exist, such as learning-based or model-based methods, with

their own advantages and drawbacks that are partially listed hereafter. If some results have shown significant results for kinematics control, few methods exist for dynamic control (Thuruthel et al. 2018*a*).

2.3.1 Open-loop

Open-loop methods provide many advantages, such as the ability to simulate the robot to validate, or not, its design, to study its controllability or to optimize the placement of sensors. Open-loop simulation is also used to estimate the end-effector workspace of a cable-driven robot in (Diao & Ma 2006) or to learn the entire workspace of a humanoid robot in(Jamone et al. 2012).

Many works are focused on open loop control of soft robots (Onal & Rus 2013, Umedachi et al. 2013). The computational cost of accurate models is a major drawback of model-based controllers, both because of the complexity of the design and for the practical implementation on the hardware. To avoid this constraint, authors of (Thuruthel et al. 2018*b*) propose a learning-based open-loop dynamic controller to perform dynamic motions. In addition, a trajectory planning method is presented in (Lismonde et al. 2017) where an inverse dynamics problem is solved.

With regard to model-based approaches, inverse kinematics experiments were conducted based on the simulation software SOFA presented above. Under quasi-static assumptions (slow-motions), inverse kinematics simulation takes as input the desired position of the robot and computes the actuation required to achieve the desired position. Open-loop inverse kinematics experiments were conducted on real physical robots (Duriez 2013) and recent work included contacts between the robots and its environment in the optimization process (Coevoet, Escande & Duriez 2017). Figure 2.6 shows examples of soft robots for manipulation and locomotion controlled via a real-time inverse problem based on FEM model.

A dynamic model of a multi-body fluidic actuated soft manipulator is presented in (Marchese et al. 2016). Authors start from the energy formulation to derive a dynamic model to be used within the iterative learning control algorithm. The formulation of the potential energy of a segment of the robot arm is defined as:

$$V_p(\theta) = mgz(\theta) \tag{2.3}$$

where θ is the bend angle, m the segment mass, g is the gravitational term and z is the position along the z-axis. This definition combined with constant curvature assumption makes it possible to write the soft robot dynamics in a traditional manipulator equation form as:

$$H(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + G(\theta) = B\tau \tag{2.4}$$



Figure 2.6: *Examples of soft robots controlled via a real-time inverse problem based on FEM model, from (Coevoet et al. 2019).*

Authors show how to use this model to derive an algorithm to perform dynamic maneuvers using locally-optimal open-loop strategies.

Finally, biology is a major source of inspiration to design soft robots. Authors of (Rozen-Levy et al. 2019) present a caterpillar inspired soft robot - see figure 2.7 - well suited to move in complex 3D environment. It uses actively controlled gripping mechanisms to crawl along branches.

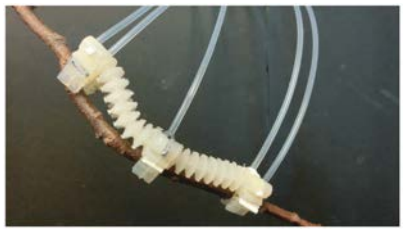


Figure 2.7: *Branch Bot: a caterpillar inspired soft robot controlled via active gripping mechanisms, from (Rozen-Levy et al. 2019).*

However, open-loop algorithms suffer from well known performance and robustness limitations. For instance, open-loop algorithms based on FEM models are sensitive to mechanical parameters, mesh discretization, boundary conditions etc. In the following, recent advances on closed-loop control of soft robots are gathered.

2.3.2 Closed-loop

Open-loop controllers do not depend on the systems states and outputs, they are also sensitive to modeling uncertainties or to external perturbations. To guarantee that the system states and/or outputs reach a desired setpoint and/or trajectory, closed-loop controllers are used.

Depending on the applications, one may study only the kinematics behavior of the robot. Kinematics is the branch of mechanics that is interested in the study of motions without considering the forces that cause them. For robotics applications, it consists in neglecting the inertial effect on the robot. However, for some applications, for instance when one wants to achieve a high speed maneuver, these effects cannot be neglected. One has to study the dynamic behavior of the robot. Dynamics is a discipline of classical mechanics that studies motion of structures under the influence of the mechanical actions that are applied to them.

To design feedback controllers for soft robot, depending if one is interested in controlling the kinematics or dynamics of the structure, it can lead to different controller design. As for open-loop algorithms, many different methods exist to tune a closed-loop controller, depending on the modeling strategy, the targeted application or an optimization criterion.

Kinematics

On the one hand, some authors opt for model-free methods to design kinematics controllers. Model-free control methods have been proposed to overcome the modeling obstacle of soft robots. Based on reinforcement learning, authors of (You et al. 2017) present a methodology that focuses on learning the control strategy rather than the physical model. Experiments on a real 2D soft manipulator are conducted to show the interest of the proposed method. In addition, authors of (Yip & Camarillo 2016) propose a hybrid position/force control method that also uses a model-less approach. A learning phase estimates the manipulator Jacobian while handling contacts with the robot environment.

On the other hand, many model-based kinematics controllers exist in the litterature. Based on a piece-wise constant curvature kinematic model, authors of (Wang et al. 2017) developed an adaptive visual control strategy for soft manipulators in constrained environments. Also based on visual servoing, (Fang et al. 2019) presents a controller constructed by learning the inverse model used to perform inverse kinematic control. The actuator input is α and s is the manipulator configuration under this input. The forward model takes the

form:

$$s(k+1) = s(k) + f(s(k), \alpha(k+1) - \alpha(k)) \tag{2.5}$$

Then, to generate the actuation command, authors propose to learn the inverse kinematics mapping using experimental data.

Based on a real-time finite element model, (Zhang et al. 2017) proposes a visual servoing control method, the robot is simulated in real time and a state observer makes sure that the configurations of both the real robot and its simulated model stay close. This method allows positional control of the robot but are restricted to slow motions due to the quasi-static model used to design the controller.

Dynamics

Kinematics controllers may not be sufficient to perform high speed tasks (such as jumping) or high speed obstacle avoidance. Recently, different authors have proposed new methods to control the dynamic behavior of soft robots. Model-free dynamic controllers applied to soft robots are not widely spread in the community. Most of the existing methods are based on a dynamic model of the deformable structure.

Authors of (Della Santina et al. 2017) highlight the challenge of controlling soft robots without reducing the natural and desired compliance of the structure. They point out the difficulty in designing a good feedback controller that preserves the mechanical behavior and guarantees desired performances. In addition, (Thuruthel et al. 2018*a*) describes the dynamic control of soft robot as *'probably the most challenging field in the control of soft robots'*. In recent years, researchers have published several contributions, some of which are gathered hereafter.

A dynamic controller based on constant curvature model is presented in (Falkenhahn et al. 2015) where the model is used to generate a feed-forward action coupled with a PID controller. Also based on piece-wise constant curvature model, authors of (Della Santina et al. 2018) present a dynamic controller that enables dynamic trajectory tracking for a continuous soft robot while handling interactions with the environment. Based on the PCC model (2.1), authors propose a feedback controller for trajectory following in the soft robots space q:

$$\tau = A^{-1}(q)(G(q)+C(q,\dot{q})\dot{\bar{q}}+M(q)\ddot{\bar{q}}+K\bar{q}+K_p(\bar{q}-q)+D(q)\dot{\bar{q}}+K_D(\dot{\bar{q}}-\dot{q})) \tag{2.6}$$

where $\bar{q}$ is the desired evolution of q in the curvature space. This leads to the following closed-loop dynamics:

$$M(q)\ddot{e} + C(q,\dot{q})\dot{e} = -(K+K_P)e - (D(q)+K_D)\dot{e} \tag{2.7}$$

where $e = \bar{q} - q$. Experiments are presented on a 2D planar soft arm and extended to 3D experiments in (Katzschmann, Della Santina, Toshimitsu, Bicchi & Rus 2019), see figure 2.8. This 3D robot is also used later in this manuscript to validate the control law based on FEM model.

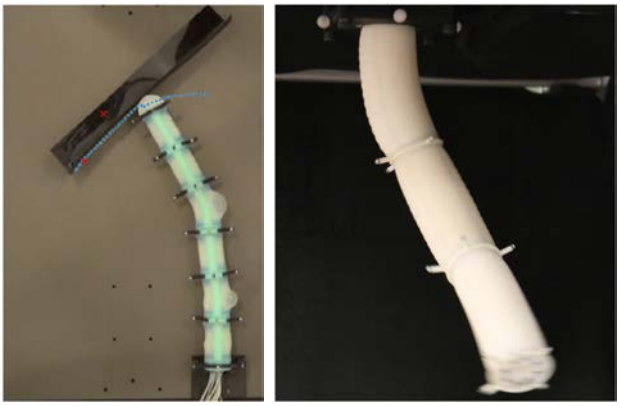


Figure 2.8: *Planar soft robot dynamically controlled based on PCC model with contact handling in (Della Santina et al. 2018) with its extension to 3D soft robot in (Katzschmann, Della Santina, Toshimitsu, Bicchi & Rus 2019).*

Combining Cosserat and Lagrange dynamic models with Ritz-Galerkin methods, (Sadati et al. 2018) presents a real-time model of continuum manipulators that enables nonlinear impedance and configuration control. In addition, model predictive control has been used recently to perform trajectory following tasks (Bruder et al. 2019). Authors use Koopman operator and system identification methods to construct a discrete-time linear model based on which a linear model predictive controller is designed. Figure 2.9 shows the experimental platform on which the control algorithms have been succesfully validated.

Soft robots are characterized by a highly under-actuated input space. Authors of (Della Santina, Pallottino, Rus & Bicchi 2019) introduce synergestic control to deal with this under-actuation issue. For now, only simulation experiments were conducted, as the design of the state observer for this control method is still an open problem.

There is a tight interplay between the physical robots and the physical world, whenever the robot moves it will affect the environment and be infludenced by it. Morphological computation tends to take over some of the processes normally attributed to control. The underlying idea is to design a simple controller but optimize the robots morphology to achieve challenging tasks and behaviors. Authors of (Hauser et al. 2012) discuss how to integrate feedback in morphological computation with soft robots. Using machine learning techniques,

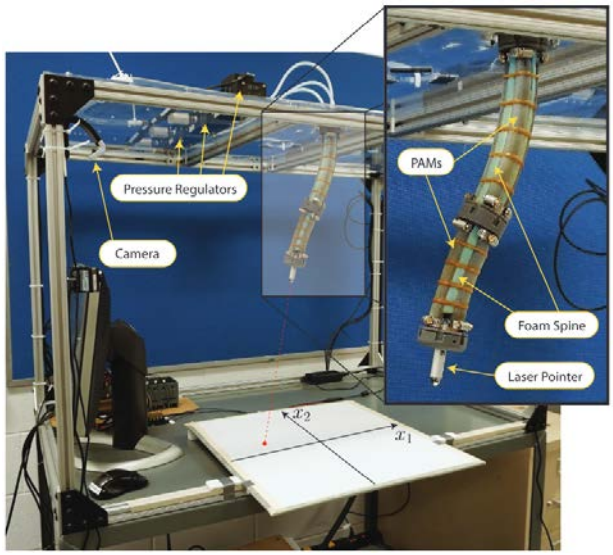


Figure 2.9: *Soft robot controlled using Koopman operator and model predictive control in (Bruder et al. 2019).*

it has been shown in (Nakajima et al. 2014) how to use the soft body dynamics as a reservoir and exploit short-term memory to control a soft arm in a closed-loop manner.

Dynamic control of soft robots using finite element method has first been studied in (Thieffry et al. 2017), where a linear finite element model coupled with model reduction algorithms is used to control the dynamic behavior of soft robots in simulation experiments. The same modeling technique is used in (Thieffry et al. 2019) where a state observer is added to the control design to enable real-world experiments using a cable-driven soft robot. This method has also been tested on a pneumatically actuated soft arm in (Katzschmann, Thieffry et al. 2019).

2.4 Some recent works about control theory

Studying soft robots raises theoretical questions that one may encounter in different fields of research such as construction of space structure (Boning & Dubowsky 2010), active vibration control (Zhang et al. 2016), study of wing span in a wind tunnel (Demourant & Poussot-Vassal 2017), or to control a crane with chains of payload (Stürzer et al. 2018). In the latter, authors study the energy flows in the system to take benefit from the mechanical structure for the controller design. A similar idea is detailed in chapter 4. Different control strategies have been employed to study the aforementioned systems, such as

flatness (Petit & Rouchon 2001) or LMIs.

In the remainder of this document, the control problems will be formulated as Linear Matrix Inequality (LMI) constraint problems. They have been used in a wide variety of applications, such as vehicle dynamics in (Nguyen et al. 2018) or for position estimation and fault detection for rail transportation system in (Aguiar et al. 2018).

In the robotics community, it has also widely been studied. Authors of (Nguyen et al. 2019) use fuzzy modeling techniques to model the nonlinear dynamics of a two degrees of freedom serial manipulator. The control goal is presented as a LMI constraints problem to achieve guaranteed $\mathcal{H}_\infty$ tracking performances. In addition, a 6 DOF robot manipulator is controlled via LMI constraints problem in (Wang & Liu 2016). A LMI-based predictive controller is built thanks to a polytopic model of a visual servoing system. Finally, authors of (Ryan & Kim 2013) present the gain synthesis for robust quadrotor control. The control is based on feedback linearization and generates suboptimal gains with respect to $\mathcal{H}_2$ and $\mathcal{H}_\infty$ performances.

A semi-definite problem (SDP) is an optimisation problem of the form:

$$\begin{aligned} &minimize \quad c^T x \\ s.t. \quad &x \in \mathbb{R}^n : F_0 + x_1 F_1 + \ldots + x_n F_n \geq 0 \end{aligned} \tag{2.8}$$

where F_i are symmetric real-valued matrices of same dimensions. The set of all the constraints of this SDP is defined by a Linear Matrix Inequality (LMI) of the form

$$F(x) = F_0 + \sum_{i=1}^{n} x_i F_i > 0 \tag{2.9}$$

where $x \in \mathbb{R}^m$ is the decision variable and the symmetric matrices $F_i \in \mathbb{R}^{n \times n}$ are known (Boyd et al. 1994).

It has been widely studied in the control theory community to study stability and robustness of systems or to design closed-loop controller (Gahinet & Apkarian 1994). LMI constraints problems are for instance used to study the stability of systems using Lyapunov theoretical framework. Let us consider a linear system of the form $\dot{x} = Ax,\ x \in \mathbb{R}^n$, the existence of a matrix P positive definite such that:

$$A^T P + PA < 0 \tag{2.10}$$

is a necessary and sufficient condition to ensure the convergence to 0 of the state whatever are its initial conditions $x(0)$.

Part II

Large-Scale Model and Controller

Introduction

This first part is dedicated to the study of the large-scale dynamic model and its control, without using model reduction. Chapter 3 presents the modeling methodologies used in this work to build a dynamic model of the robot studied. This modeling step is based on the Finite Element Method (FEM), a numerical method widely used in numerical analysis of partial differential equations (Cook et al. 2007, Reddy 1993).

Among others modeling strategies, like constant curvature models, this modeling method aims at being generic regarding the geometry of the robot. It is not shape specific and is directly usable on any soft robot, if a spatial mesh of its structure is provided. The mesh generation is an independant research topic and we rely on existing software to obtain the structure mesh. The SOFA framework embeds for instance the CGAL[2] plugin to generate a mesh from the geometry of the structure (i.e. a CAD file). The modeling procedure, relying on the SOFA framework, is also fully automatised (Coevoet et al. 2017).

The more nodes the FEM model has, the more it tends to be accurate, a precise mesh is often made of thousands of elements. The associated dynamic model is also made of thousands of state variables, yielding to large-scale dynamical systems. This dimensionality issues bring challenges during several steps of the design of the model and controller.

First, it is challenging to reach real-time performances with a precise simulation. Recent work has been done to speed up the simulation using GPU methods (Allard et al. 2012). Handling contacts between the robot and its environments brings additionnal constraints to the simulation, this is an active research topic that will not be detailed in this thesis but solutions are included into the SOFA framework to deal with these issues (Coevoet, Escande & Duriez 2017).

This dimension issue also brings challenges from a control point of view. Nowadays, it is indeed not possible to use off-the-shell control design tools to compute a dynamic controller for a system with these dimensions. We have to derive new

[2]Computational Geometry Algorithms Library, https://www.cgal.org/index.html

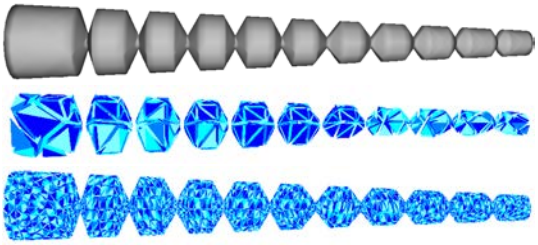


Figure 2.10: *Examples of spatial discretization for structure of the trunk robot presented in figure of 3.1;*
Top: Visual model; Center: Coarse mesh; Bottom: medium-size mesh used in this work.

paradigms to design a control for this kind of system, in this work we assume a stable open-loop behavior and take profit of this property to use an open-loop Lyapunov function. Under pure elastic behavior the robot goes back to its initial position when the actuation vanishes. Under this assumption, chapter 4 presents a control design for large-scale systems with simulation experiments.

Chapter 3

Finite Element Model

This chapter presents the model on which the control law is built. Modeling soft robots implies to solve equations coming from continuum mechanics. To overcome the difficulty of studying an analytical model, we rely on numerical methods to solve the equations. In this work, we rely on the finite element method to build the model, it consists of discretizing the structure into small elements, the equations of continuum mechanics are then solved for each of these elements. This modeling strategy is implemented using the SOFA framework (Allard et al. 2007) and the soft robots plugin, see (Coevoet et al. 2017) for details.

Contents

3.1 Nonlinear second order model 30
3.2 Nonlinear state-space equation 30
3.3 Linear large-scale state-space equation 32
3.3.1 Continuous-time linear state-space model 33
3.3.2 Discrete-time linear state-space model 34
3.4 Illustrations 35
3.4.1 Trunk like robot 35
3.4.2 Pressurized 3D arm 37

3.1 Nonlinear second order model

Let us start with the formulation given by the second law of Newton (see (Coevoet et al. 2017) for details), that models the dynamic behavior of a body as:

$$\mathbf{M}(q)\ddot{q} = \mathbf{P}(q) - \mathbf{F}(q, \dot{q}) + \mathbf{H}^T(q)\lambda \tag{3.1}$$

where $q \in \mathbb{R}^n$ is the vector of generalized degrees of freedom, $\mathbf{M}(q) : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ is the inertia matrix, $\mathbf{F}(q, \dot{q}) : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ represents the internal forces applied to the structure and $\mathbf{P}(q) : \mathbb{R}^n \to \mathbb{R}^n$ gathers known external forces. Finally, matrix $\mathbf{H}(q) : \mathbb{R}^m \to \mathbb{R}^{n\times m}$ is the matrix containing the actuation directions while $\lambda \in \mathbb{R}^m$ is the vector of actuators forces.

Different laws exist to model the strain-stress relationship of deformable materials. The Hooke's law is one of these laws, it assumes a linear ratio between the material response and the loads. Different constitutive laws have different computational costs. Under pure elastic assumption, i.e. the robot goes back to its rest shape when the actuation vanishes, the parameters are given by Young's modulus and Poisson's ratio of Hooke's law.

In equation (3.1), the vector q is the displacement of each nodes of the mesh in the three dimensions of space x, y and z. It is also defined as:

$$q = \begin{pmatrix} q_{x_0} & q_{y_0} & q_{z_0} & q_{x_1} & q_{y_1} & q_{z_1} & \dots & q_{x_\mathcal{N}} & q_{y_\mathcal{N}} & q_{z_\mathcal{N}} \end{pmatrix}^T \tag{3.2}$$

Let $\mathcal{N}$ be the number of nodes of the mesh, the number of variables in q is $n = 3 \times \mathcal{N}$. The more precise the model is, the more variables there are in the model. Let us define the vector of velocity $v \in \mathbb{R}^n$, $v = \dot{q}$, the state $x \in \mathbb{R}^{2n}$ of the robot is then defined as:

$$x = \begin{pmatrix} v \\ q \end{pmatrix} \tag{3.3}$$

The following section details the definition of both continuous time and discrete time state-space equations.

3.2 Nonlinear state-space equation

The internal force vector $\mathbf{F}$ is a nonlinear function of the positions and the velocities. We then apply a Taylor series expansion to $\mathbf{F}$ and make the following first order approximation:

$$\mathbf{F}\ (q + \delta q, v + \delta v) \approx \mathbf{F}(q, v) + \frac{\partial \mathbf{F}(q, v)}{\partial q}\delta q + \mathbf{D}(q, v)\delta v \tag{3.4}$$

We define the stiffness matrix $\mathbf{K}$ and the damping matrix $\mathbf{D}$ as:

$$\begin{aligned} \mathbf{K} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^{n\times n} \quad \mathbf{K}(q,v) &= \frac{\partial \mathbf{F}(q,v)}{\partial q} \\ \mathbf{D} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^{n\times n} \quad \mathbf{D}(q,v) &= \frac{\partial \mathbf{F}(q,v)}{\partial v} \end{aligned} \tag{3.5}$$

From equation (3.1), the non-linear continuous-time model directly writes:

$$\begin{pmatrix} \dot{v} \\ \dot{q} \end{pmatrix} = \begin{pmatrix} \mathbf{M}^{-1}(q)[\mathbf{P}(q) - \mathbf{F}(q,v)] \\ v \end{pmatrix} + \begin{pmatrix} \mathbf{M}^{-1}(q)\mathbf{H}^T(q) \\ 0 \end{pmatrix} \lambda \tag{3.6}$$

The continuous-time state-space equation is straightforward, provided that the mass matrix $\mathbf{M}$ is regular. However, the simulation software used to model the robot implements a discrete-time version of equation (3.1) that is presented hereafter.

The simplest method to numerically solve the mechanical equation is the Euler explicit method. However, this method presents a major drawback as the time step used in the simulation is constrained by the length of the smallest element of the mesh. To have unconditional numerical stability, we integrate equation (3.1) using a time-stepping implicit scheme (backward Euler).

Let us consider the time interval $[t_k, t_{k+1}]$ whose length is $h = t_{k+1} - t_k$, and we denote $\delta q = q_+ - q = hv_+$ and $\delta v = v_+ - v$; with these notations, equation (3.1) writes:

$$\begin{aligned} \mathbf{M}(q_+)\delta v &= h\Big(\mathbf{P}(q_+) - \mathbf{F}(q_+, v_+)\Big) + h\mathbf{H}^T(q_+)\lambda \\ q_+ &= q + hv_+ \end{aligned} \tag{3.7}$$

From (3.7) and (3.4), the equation of motion writes:

$$\mathbf{M}(q_+)\delta v = h\Big(\mathbf{P}(q_+) - \mathbf{F}(q,v) - \mathbf{K}(q,v)\delta q - \mathbf{D}(q,v)\delta v\Big) + h\mathbf{H}^T(q_+)\lambda \tag{3.8}$$

As $\delta q = hv_+ = h\delta v + hv$, it follows:

$$\begin{aligned} \mathbf{M}(q_+)\delta v =& h\mathbf{P}(q_+) - h\mathbf{F}(q,v) - h\mathbf{K}(q,v)(h\delta v + hv) - h\mathbf{D}(q,v)\delta v \\ &+ h\mathbf{H}^T(q_+)\lambda \end{aligned}$$

$$\Leftrightarrow$$

$$\begin{aligned} \Big(\mathbf{M}(q_+) + h^2\mathbf{K}(q,v) + h\mathbf{D}(q,v)\Big)\delta v =& h\mathbf{P}(q_+) - h\mathbf{F}(q,v) - h^2\mathbf{K}(q,v)v \\ &+ h\mathbf{H}^T(q_+)\lambda \end{aligned} \tag{3.9}$$

Defining

$$\mathbf{S}(q, q_+, v) = \Big(\mathbf{M}(q_+) + h^2\mathbf{K}(q, v) + h\mathbf{D}(q, v)\Big) \tag{3.10}$$

it follows:

$$\mathbf{S}(q, q_+, v)\delta v = h\bigg(\mathbf{P}(q_+) - \mathbf{F}(q, v) - h\mathbf{K}(q, v)v\bigg) + h\mathbf{H}^T(q_+)\lambda \tag{3.11}$$

As $\delta v = v_+ - v$, the equation of motion finally writes:

$$\begin{aligned} v_+ =& v + h\mathbf{S}(q, q_+, v)^{-1}\bigg(\mathbf{P}(q_+) - \mathbf{F}(q, v) - h\mathbf{K}(q, v)v\bigg) + h\mathbf{S}(q, q_+, v)^{-1}\mathbf{H}^T(q_+)\lambda \\ q_+ =& q + hv_+ \end{aligned} \tag{3.12}$$

Considering the state vector $x = \begin{pmatrix} v & q \end{pmatrix}^T$, the non-linear discrete-time state-space equation is:

$$\begin{aligned} \begin{pmatrix} I & 0 \\ -hI & I \end{pmatrix} x_+ = & \begin{pmatrix} v + h\mathbf{S}(q, q_+, v)^{-1}\Big(\mathbf{P}(q_+) - \mathbf{F}(q, v) - h\mathbf{K}(q, v)v\Big) \\ q \end{pmatrix} \\ & + \begin{pmatrix} h\mathbf{S}(q, q_+, v)^{-1}\mathbf{H}^T(q) \\ 0 \end{pmatrix} \lambda \end{aligned} \tag{3.13}$$

In the remainder of this manuscript, controllers are designed based on a linearized model, presented in the next section.

3.3 Linear large-scale state-space equation

Let $q_0 \in \mathbb{R}^n$ be a stable equilibrium point. It is induced by the gravity field $\mathbf{P}(q_0)$ and the actuation input λ_0.

The Taylor expansion is made around this configuration q_0 and it holds:

$$\begin{aligned} \mathbf{F}(q, v) &\approx \mathbf{F}(q_0, 0) + \left.\frac{\partial \mathbf{F}(q, v)}{\partial q}\right|_{\substack{q=q_0 \\ v=0}} \partial q + \left.\frac{\partial \mathbf{F}(q, v)}{\partial v}\right|_{\substack{q=q_0 \\ v=0}} \partial v \\ &= \mathbf{F}(q_0, 0) + \mathbf{K}(q_0)(q - q_0) + \mathbf{D}(q_0, 0)v \end{aligned} \tag{3.14}$$

Assumption 1. *Hereafter, we assume that the external forces $\mathbf{P}(q)$, the mass matrix $\mathbf{M}(q)$, the stiffness $\mathbf{K}(q, v)$, the damping $\mathbf{D}(q, v)$ and the actuation*

direction $\mathbf{H}(q)$ *are constant over time, i.e. the following holds:*

$$\begin{aligned} \mathbf{P} &= \mathbf{P}(q) = \mathbf{P}(q_0) \\ \mathbf{H} &= \mathbf{H}(q) = \mathbf{H}(q_0) \\ \mathbf{M} &= \mathbf{M}(q) = \mathbf{M}(q_0) \\ \mathbf{K} &= \mathbf{K}(q, v) = \mathbf{K}(q_0, 0) \\ \mathbf{D} &= \mathbf{D}(q, v) = \mathbf{D}(q_0, 0) \end{aligned} \tag{3.15}$$

Rayleigh damping is commonly used in solving finite element analysis (Mohammad et al. 1995, Hall 2006). It provides a source of energy dissipation in the analysis of mechanical systems. The Rayleigh damping matrix consists of a mass-proportional and a stiffness-proportional part, the matrix $\mathbf{D}$ is therefore defined as:

$$\mathbf{D} = \alpha\mathbf{M} + \beta\mathbf{K} \tag{3.16}$$

where α and β are respectively the mass and stiffness-proportional damping parameters. The Rayleigh damping is a numerical artefact and has therefore no physical meaning.

Finally, we define the displacement vector d as

$$d = q - q_0 \tag{3.17}$$

3.3.1 Continuous-time linear state-space model

As q_0 is an equilibrium point, it holds:

$$0 = \mathbf{P} - \mathbf{F}(q_0, 0) + \mathbf{H}^T\lambda_0 \tag{3.18}$$

From (3.1) and (3.14), the equation of motion around this equilibrium point q_0 writes:

$$\mathbf{M}\dot{v} = \mathbf{P} - \mathbf{F}(q_0, 0) - \mathbf{K}d - \mathbf{D}v + \mathbf{H}^T\lambda \tag{3.19}$$

Computing (3.19)-(3.18) yields to the following equation, modeling the motion around an equilibrium point q_0:

$$\mathbf{M}\dot{v} = -\mathbf{K}d - \mathbf{D}v + \mathbf{H}^T(\lambda - \lambda_0) \tag{3.20}$$

Under this assumption and defining a new input $u = \lambda - \lambda_0$, we obtain a linear model of the behaviour of the robot described by the following equation:

$$\mathbf{M}\dot{v} = -\mathbf{K}d - \mathbf{D}v + \mathbf{H}^T u \tag{3.21}$$

Considering the state vector $x = \begin{pmatrix} v & d \end{pmatrix}^T$, equation (3.21) can be written as a Linear Time Invariant (LTI) model:

$$\begin{cases} \dot{x} = \begin{pmatrix} -\mathbf{M}^{-1}\mathbf{D} & -\mathbf{M}^{-1}\mathbf{K} \\ I & 0 \end{pmatrix} x + \begin{pmatrix} -\mathbf{M}^{-1}\mathbf{H}^T \\ 0 \end{pmatrix} u \\ y = Cx \end{cases} \tag{3.22}$$

where I is the identity matrix of dimension n and the matrix C is a sparse matrix defining the end- effector coordinates.

3.3.2 Discrete-time linear state-space model

From (3.13) and (3.15) and defining the new input vector $u = \lambda - \lambda_0$ and state vector $\mathbf{q} = q - q_0$, $x = (v \quad \mathbf{q})^T$, the linear state space equation is:

$$\begin{pmatrix} I & 0 \\ -hI & I \end{pmatrix} x_+ = \begin{pmatrix} v - h\mathbf{S}^{-1}\Big[\mathbf{F}(q,v) + h\mathbf{K}v\Big] \\ \mathbf{q} \end{pmatrix} + \begin{pmatrix} h\mathbf{S}^{-1}\mathbf{H}^T \\ 0 \end{pmatrix} u \tag{3.23}$$

Time step h is a scalar, thus we have:

$$\begin{pmatrix} I & 0 \\ -hI & I \end{pmatrix}^{-1} = \begin{pmatrix} I & 0 \\ hI & I \end{pmatrix} \tag{3.24}$$

From (3.13) to (3.23), $\mathbf{P}$ vanishes as it is included in the equilibrium point. Finally, $\mathbf{F}(q,v)$ are the internal forces that are approximated around the equilibrium q_0 by equation (3.14), i.e. $\mathbf{F}(q,v) = \mathbf{F}(q_0,0) + \mathbf{K}\mathbf{q} + \mathbf{D}v$. The constant part $\mathbf{F}(q_0,0)$ is included in the equilibrium point and equation (3.23) writes:

$$x_+ = \begin{pmatrix} I & 0 \\ hI & I \end{pmatrix} \begin{pmatrix} v - h\mathbf{S}^{-1}\Big[\mathbf{K}\mathbf{q} + \mathbf{D}v + h\mathbf{K}v\Big] \\ \mathbf{q} \end{pmatrix} + \begin{pmatrix} I & 0 \\ hI & I \end{pmatrix} \begin{pmatrix} h\mathbf{S}^{-1}\mathbf{H}^T \\ 0 \end{pmatrix} u \tag{3.25}$$

Finally, the state space equation can be written in standard form:

$$M \;:\; \begin{cases} x_+ = \underbrace{\begin{pmatrix} I - h\mathbf{S}^{-1}(\mathbf{D} + h\mathbf{K}) & -h\mathbf{S}^{-1}\mathbf{K} \\ hI - h^2\mathbf{S}^{-1}(\mathbf{D} + h\mathbf{K})) & I - h^2\mathbf{S}^{-1}\mathbf{K} \end{pmatrix}}_{A} \begin{pmatrix} v \\ \mathbf{q} \end{pmatrix} + \underbrace{\begin{pmatrix} h\mathbf{S}^{-1}\mathbf{H}^T \\ h^2\mathbf{S}^{-1}\mathbf{H}^T \end{pmatrix}}_{B} u \\ y = Cx \end{cases} \tag{3.26}$$

where $A \in \mathbb{R}^{2n\times 2n}$, $B \in \mathbb{R}^{2n\times m}$, m being the number of actuators and $C \in \mathbb{R}^{p\times 2n}$ is a matrix defining the end effector coordinates and p is the number of outputs.

3.4 Illustrations

In this work, the theoretical results are illustrated thanks to two different soft robots whose information are gathered in table 3.1. More details are given thereafter.

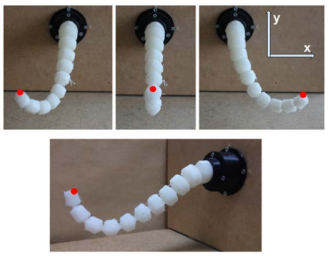
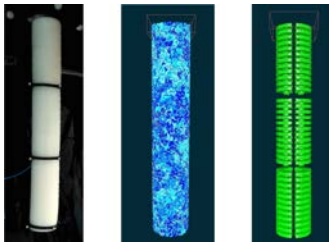
	Trunk	Pressurized soft arm
		
Actuation:	Cables	Pressure
Materials:	Silicone	Silicone
Length (cm):	18	33
Diameter (cm):	3 (base) to 1 (tip)	4.5
# nodes in model:	1557	11810

Table 3.1: *Soft robots used to validate the theoretical results*

3.4.1 Trunk like robot

This experimental platform is a trunk-like robot presented in figure 3.1 with a schematic view in figure 3.2. It is made entirely of silicone, it is 18 centimeters long and the thickness at its base and its tip are respectively 2.5 and 1 centimeters. The structure is driven by 4 cables - actuated by 4 servomotors whose entry are the cable length - that permit to work in the 3 dimensions of space. The output of the system is the position of the tip (red point in figure 3.1) that is measured using a magnetic micro-sensor whose frequency can reach $240Hz$.

Dynamic FEM model

A comparison of different FEM mesh with different accuracy is given in figure 3.3. A mesh with 210 nodes is not accurate enough to represent the geometry faithfully. Conversely, a mesh with 6012 nodes does not give more accurate results compared to a medium size mesh of 1557 elements. This medium size mesh is a good compromise and will be used in this work.

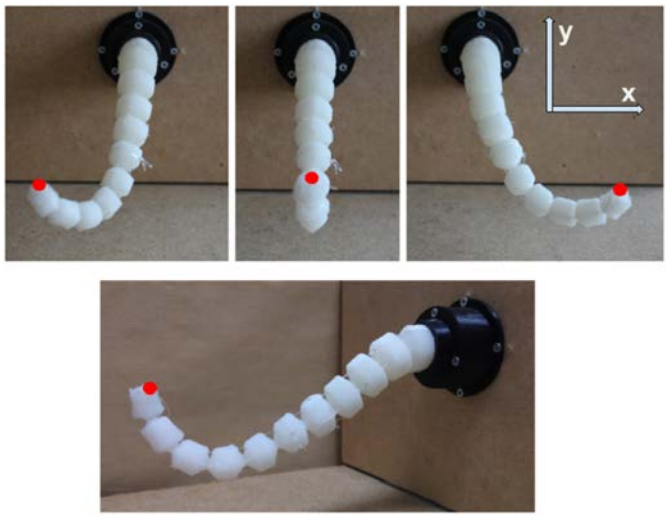


Figure 3.1: *Soft robot used for experimental validation, fully made of silicone. Top: front view of the robot. Bottom: side view. Red = end-effector, location of the sensor.*

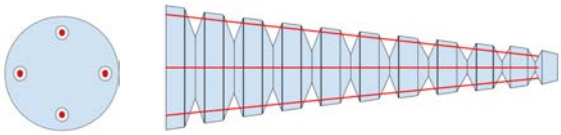


Figure 3.2: *Design of the robot: slice view (left) and side view (right). The robot is actuated with* 4 *cables in red.*

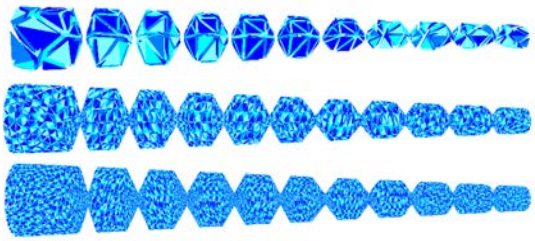


Figure 3.3: *Comparison of different FEM mesh of the Trunk-like robot with different accuracy.*
Top: FEM mesh with 210 *nodes, Middle: FEM mesh with* 1557 *nodes, Bottom: mesh with* 6012 *nodes.*

The finite-element mesh of this robot is made of 1557 nodes. The dimensions of the state vector in system (3.22) and (3.26) is also $1557 \times 3 \times 2 = 9342$ state variables (3 directions of space for displacement and velocity), as shown in figure 3.4. The output is the position of the end-effector (red-point of figure 3.1) in the two directions of space. A comparison of the real system behavior

and the simulated model is given in figure 3.5, that shows a comparison of the step response of the real robot and its simulated model. Experiments show that the main behavior of the robot is captured by the model as long as the robot stays inside the region where the linearization assumption is valid.

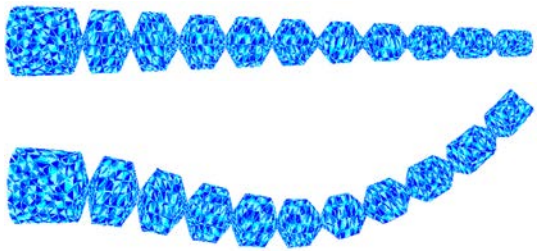


Figure 3.4: *FEM mesh of the Trunk-like robot presented in figure 3.1. The mesh is made of* 1557 *nodes and* 2972 *tetrahedrons.*
Top: rest position, Bottom: deformed position

Figure 3.5: *Comparison of the real robot and dynamic model.*
Top / Bottom: output along x and y axis.
Left: Step response of the real robot,
Right: Step response of the simulated model.

Remark: With an accurate calibration process, it is possible to obtain a zero static error in open-loop. However, the open-loop algorithm does not offer any guarantee about the performances of the control algorithm, especially in presence of perturbations. Moreover, due to hardware limitations, the calibration has to be done regularly as phenomena such as fatigue are difficult to model.

3.4.2 Pressurized 3D arm

This robot was built in the Distributed Robotics Laboratory, inside the Computer Science and Artificial Intelligence Lab (CSAIL) at MIT. Its design is inspired from previous work (Marchese et al. 2015, Marchese & Rus 2016).

This soft robot is composed of three segments with four inflatable cavities per segment. Each segment of the soft arm is 11cm long and has a diameter of 4.5cm. The geometry is detailed in Figure 3.6 while the manufacturing steps are outlined in Figure 3.7.

This soft robot is a concatenation of cylindrical soft actuator segments. Each segment has inflatable cavities shaped as a series of thin ribs. The arm is composed of three segments with four inflatable cavities in each segment. Each cavity has a ribbed interior geometry that allows for more bending and less radial inflation when compared to a purely cylindrical cavity design. Each ribbed cavity consists of 16 ribs connected by a thin connecting channel between those 16 ribs. Each rib is 3.1mm high and between each rib is a gap of 3mm. The thickness between each rib and the outer surface of a segment is only 2mm.

Figure 3.6: *Geometric model of the soft arm. From left to right: the shape of a single ribbed cavity; circular arrangement of four cavities; soft arm segment with embedded cavities; concatenation of three actuated segments plus passive base.*

The four wax cores are created through injecting liquefied bleached bees-wax into a rubber mold. The wax cores are removed from the mold, post-processed to remove any unnecessary residue, and then assembled into a 3D printed mold for casting a single arm segment. Silicone Rubber is mixed, degassed and filled into the mold. The mold is disassembled and the resulting segment is placed in an oven to melt out the wax and afterwards cooked under boiling water to remove any wax residue. Silicone tubing is then glued into one side and silicone rod stock is used to close up the other end of the arm segment. Two more arm segments are manufactured in the same manner. Finally, all silicone tubing is routed so that all three segments can be properly concatenated and glued together. Finally, all tubing is labeled and motion capture markers are added.

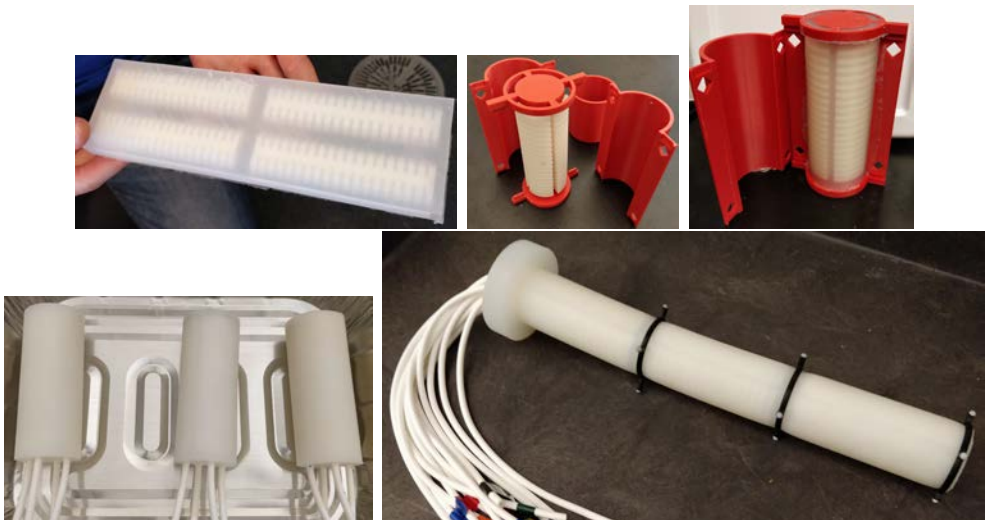


Figure 3.7: *Manufacturing overview of the 3D soft arm. From Top Left to Bottom Right: Creation of four wax cores through casting; assembly of wax cores into 3D printed molds; casting of Silicone Rubber into the mold; Removal of wax through melting and addition of silicone tubing; Routing of tubing, gluing, labeling and adding markers.*

Actuation and Motion Capture

The independent pneumatic actuation of the arm segments is achieved through an array of 12 pressure-controlled proportional valves. A motion tracking system provides real-time measurements of marked points along the in-extensible back of the soft arm. A rigid frame holds all the sub-systems together providing reliable hardware experiments without the need for re-calibration of the infrared cameras of the motion capture system.

Dynamic FEM model

To build a precise simulation, elastic and inertial parameters have to be tuned in simulation. The Young's modulus of the structure is directly obtained from silicone's properties and the mass of the arm is measured experimentally. The Rayleigh damping parameters are then tuned experimentally.

The mesh of the soft arm is presented in figures 3.8. From our testing, the mesh that gives the better tradeoff beetween accuracy and number of nodes consists of 45116 tetrahedra and 11810 nodes. The state vector of the dynamic model has a length of $11810 \times 6 = 70860$. The model implemented in SOFA is non-linear, but in order to design a dynamic controller, the model of the arm is linearized around its resting position.

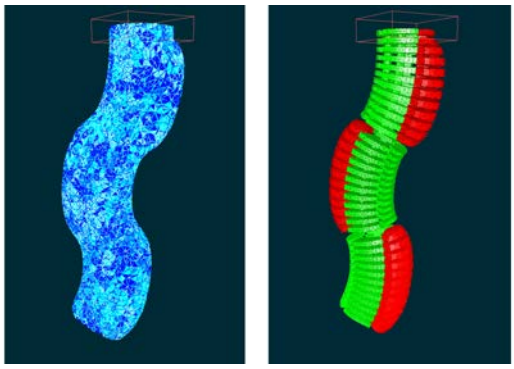


Figure 3.8: *Example of deformation state of the soft arm in simulation.*
Left: tetrahedral mesh, Right: mesh of the inner cavites: in red, actuated cavities.

Figure 3.9: *Open-loop arm experiment*
Left: starting from an actuated and deformed state going towards a straight, down-hanging arm position.
Right: starting from a straight, down-hanging arm pose going towards a deformed state through an open-loop step actuation.
The graph shows the x (red), y (blue) and z (yellow) positions of the arm's end-effector as it is swinging back and forth while asymptotically nearing a straight arm position.

We validate the proposed model by comparing the open-loop behaviour of the simulation to the physical prototype. In the first experiment we release the arm from a deformed shape and let it converge to its resting position, hanging straight down. Then, a second experiment is done to study the step response of the system (i.e. the robot starts from its rest shape and converges to a deformed position); results are shown in figure 3.9. Figure 3.10 shows the output that presents the maximum absolute error between the simulated model and the real soft arm during the open-loop experiment where the robot goes from an actuated state back to its rest shape. A maximum error of about $3cm$ exists between the simulated results and the real measurements. This corresponds to a relative error of 9.1% in regards to the robot's characteristic length. During these experiments, the robot stays inside the region where the linearization

assumption is valid, the modeling error is small enough to consider the linear model accurate. The bigger the amplitude of the step signal, the larger the deformation and the larger the modeling error is.

Figure 3.10: *Maximum error between measurement outputs of simulation and physical experiment for the open-loop experiment where the robot returns from an actuated state back to its straight rest shape.*

Chapter 4

Large Scale Feedback Controller

This chapter aims at providing a controller for the full order model. For a state feedback controller, the design problem consists of finding a matrix L for the control law $u = -Lx$ such that the systems studied are asymptotically stable with desired performances. As seen in chapter 3, the state vector is of large dimensions, $x \in \mathbb{R}^n$, so is the matrix $L \in \mathbb{R}^{n \times m}$, where m is the number of inputs. The computation of the matrix L also involves a huge number of decision variables.

Let us take the example of the cable driven trunk-like robot presented in figure 3.1. It is actuated with 4 cables and its finite element mesh is made of 1557 nodes. The design of a feedback controller $u = -Lx$ would imply the computation of a matrix $L \in \mathbb{R}^{9342 \times 4}$, made of 37 368 variables. With its mesh made of 11 810 nodes and its 12 actuators, the design of a controller for the pressurized robot of figure 3.8 implies $70\ 860 \times 12 = 850\ 320$ decision variables. Using the Lyapunov framework to study the stability of these systems would then require the computation of a Lyapunov matrix $P \in \mathbb{R}^{9342 \times 9342}$ (P is a symmetric matrix and contains $\frac{n(n+1)}{2} = 43\ 641\ 153$ variables) and $P \in \mathbb{R}^{70\ 860 \times 70\ 860}$ (2 510 605 230 variables). To store this last matrix on MATLAB, it requires 18 GB of memory, that exceeds the memory of modern computers.[1]

This makes the design of a large-scale controller challenging. This chapter proposes a controller for the large-scale system using the knowledge of an

[1] Tested on an Intel Core i7 CPU 2.70GHz x 8 with 15,6 GiB RAM.

open-loop Lyapunov function, derived from the energy of the soft structure. A part of these results are presented in (Thieffry et al. 2018, 2019).

Contents

4.1 Theoretical basis 44
4.2 Energy-based Lyapunov function 45
4.3 Parameterized Energy Function 46
4.4 Simulation results 48

4.1 Theoretical basis

In (Jurdjevic & Quinn 1978), authors present a method to design asymptotically stabilizing control laws, that have been generalized in different papers as in (Faubourg & Pomet 2000). In few words, this method requires the knowledge of an open-loop Lyapunov function to derive the controller.

Consider the continuous time linear system:

$$\dot{x} = Ax + Bu \tag{4.1}$$

Let us assume that this sytem satisfies the Jurdjevic-Quinn conditions.

Jurjedvic-Quinn conditions (Linear case of Assumption 3.1 of (Faubourg & Pomet 2000)):

We assume that a function $V_0 = (x^T P x) : \mathbb{R}^n \to \mathbb{R}$ *is known and has the following properties: it is smooth, positive definite and radially unbounded; it satisfies:*

$$A^T P + PA \leq 0 \tag{4.2}$$

and

$$rk(A^T P + PA, B, AB, \ldots, A^{n-1}B) = n \tag{4.3}$$

Proposition 3.3 of (Faubourg & Pomet 2000) (Linear case):

The linear state-feedback

$$u = -B^T P x \tag{4.4}$$

makes the origin of the closed-loop system (4.5) *asymptotically stable.*

$$\dot{x} = (A - BB^T P)x \tag{4.5}$$

Remark: *In (Faubourg & Pomet 2000), authors describe this control strategy for a non-linear control affine model; similar method could also be used on the non-linear large-scale system.*

This method also provides a straightforward method to design a controller, as long as a Lyapunov function is known for the open-loop system. This is the key point and the main challenge of the method as the computation of a Lyapunov function for systems with thousands of nodes is an open issue.

Coming back to our applications, a purely elastic behavior is assumed for the robots model. Without actuation, the robot goes back to its rest position: therefore, it is assumed to be open-loop stable. The energy of the robot is thus a Lyapunov candidate function and we propose to use the knowledge of this function to derive a large-scale state feedback controller.

4.2 Energy-based Lyapunov function

The kinetic energy of a soft structure is defined as:

$$\mathscr{E}_k(v) = \frac{1}{2} v^T \mathbf{M} v \tag{4.6}$$

where $\mathbf{M}$ is the inertia matrix of the system. The potential energy is defined as:

$$\mathscr{E}_p(d) = \frac{1}{2} d^T \mathbf{K} d \tag{4.7}$$

where $\mathbf{K}$ is the stiffness matrix of the system. The total energy is also defined as:

$$\mathscr{E}(d, v) = \frac{1}{2} \begin{pmatrix} v \\ d \end{pmatrix}^T \begin{pmatrix} \mathbf{M} & 0 \\ 0 & \mathbf{K} \end{pmatrix} \begin{pmatrix} v \\ d \end{pmatrix} \tag{4.8}$$

As we have $\mathbf{M} > 0$ and $\mathbf{K} > 0$, this energy function is positive definite. It holds:

$$\mathscr{E}(d, v) > 0 \tag{4.9}$$

and

$$\mathscr{E}(d, v) = 0 \Leftrightarrow (d, v) = (0, 0) \tag{4.10}$$

The Lyapunov candidate function is:

$$V(x) = \begin{pmatrix} v \\ d \end{pmatrix}^T \begin{pmatrix} \mathbf{M} & 0 \\ 0 & \mathbf{K} \end{pmatrix} \begin{pmatrix} v \\ d \end{pmatrix} \tag{4.11}$$

The derivative of the Lyapunov function along the system trajetories writes :

$$\begin{aligned}\dot{V}(x) &= x^T\Big(A^TP + PA\Big)x \\ &= x^T\left((*) + \begin{pmatrix}\mathbf{M} & 0\\ 0 & \mathbf{K}\end{pmatrix}\begin{pmatrix}-\mathbf{M}^{-1}\mathbf{D} & -\mathbf{M}^{-1}\mathbf{K}\\ I & 0\end{pmatrix}\right)x \\ &= x^T\left((*) + \begin{pmatrix}-\mathbf{D} & -\mathbf{K}\\ \mathbf{K} & 0\end{pmatrix}\right)x \\ &= x^T\begin{pmatrix}-2\mathbf{D} & 0\\ 0 & 0\end{pmatrix}x \\ &= -2v^TDv\end{aligned} \tag{4.12}$$

The derivative of the Lyapunov function depends only of one element of the state vector, the speed, but not on the second one, the position. $\dot{V}(x)$ is therefore only negative semi-definite, which makes the origin stable, but not asymptotically. Next subsection presents tracks to extend this first result.

4.3 Parameterized Energy Function

In order to obtain asymptotic stability, we study a Lyapunov function, which can be seen as an extension of the energy definition.

$$V(x) = \begin{pmatrix}v\\ d\end{pmatrix}^T\begin{pmatrix}\mathbf{M} + (V_1 + V_1^T) & V_{12}\\ V_{12}^T & \mathbf{K} + (V_2 + V_2^T)\end{pmatrix}\begin{pmatrix}v\\ d\end{pmatrix} \tag{4.13}$$

Of course V_1, V_2 and V_{12} have to be defined such as $V(x) > 0$. Notice that this general form (4.13) introduces extra variables V_1, V_2 and V_{12} in the Lyapunov function that in complexity is equivalent to use a full matrix P, therefore, it contains too many decision variables for actual LMI solvers. In order to reduce this complexity, a good compromise has been found by fixing the following extra variables to:

$$\begin{aligned}&V_1 = V_{12} = \varepsilon\mathbf{M}\ ,\ \mathbf{M} = \mathbf{M}^T > 0, \mathbf{M} \in \mathbb{R}^{n\times n}\\ &V_2 = \varepsilon\mathbf{K} + \varepsilon\mathbf{D}\ ,\ V_2 = V_2^T > 0\ ,\ V_2 \in \mathbb{R}^{n\times n}\end{aligned} \tag{4.14}$$

and with $\varepsilon \in \mathbb{R}$. We now consider the following Lyapunov function :

$$V(x) = \begin{pmatrix}v\\ d\end{pmatrix}^T\begin{pmatrix}(1+\varepsilon)\mathbf{M} & \varepsilon\mathbf{M}\\ \varepsilon\mathbf{M} & (1+\varepsilon)\mathbf{K} + \varepsilon\mathbf{D}\end{pmatrix}\begin{pmatrix}v\\ d\end{pmatrix} \tag{4.15}$$

Parameter ε must be chosen so that $V(x)$ is a Lyapunov function in open-loop.

Lemma 1 (Thieffry et al. 2018): *Condition (4.16) is a sufficient condition for $V(x)$ defined in (4.15) to be a Lyapunov function in open-loop for system (3.22):*

$$0 < \varepsilon < \frac{\alpha}{1-\alpha} \tag{4.16}$$

where $\alpha \in \mathbb{R}^+$ is the mass-proportionnal damping parameter.

Proof:

- First, $V(x)$ has to be positive definite. Using Shur's complement on (4.15) yields to:

$$V(x) > 0 \Leftrightarrow (1+\varepsilon)\mathbf{M} > 0 \text{ and } (1+\varepsilon)\mathbf{K} + \varepsilon\mathbf{D} - \frac{\varepsilon^2}{1+\varepsilon}\mathbf{M} > 0 \tag{4.17}$$

  By definition, matrices $\mathbf{M}, \mathbf{K}$ and $\mathbf{D} = \alpha\mathbf{M} + \beta\mathbf{K}$ are positive definite. We thus have $\varepsilon > -1 \Rightarrow (1+\varepsilon)\mathbf{M} > 0$ and $(1+\varepsilon)\mathbf{K} > 0$. Thus, it holds:

$$V(x) > 0 \Leftarrow \varepsilon > -1 \text{ and } (1+\varepsilon+\varepsilon\beta)\mathbf{K} + (\varepsilon\alpha - \frac{\varepsilon^2}{1+\varepsilon})\mathbf{M} > 0 \tag{4.18}$$

  Rayleigh's damping coeffients α and β are positive scalars. A sufficient condition for $(1+\varepsilon+\varepsilon\beta)\mathbf{K} > 0$ to holds is $\varepsilon > 0$. We thus have:

$$V(x) > 0 \Leftarrow \varepsilon > 0 \text{ and } (\varepsilon\alpha - \frac{\varepsilon^2}{1+\varepsilon})\mathbf{M} > 0 \tag{4.19}$$

  In addition, it holds:

$$\begin{aligned} \varepsilon > 0 \text{ and } (\varepsilon\alpha - \frac{\varepsilon^2}{1+\varepsilon})\mathbf{M} &> 0 \\ \Leftarrow \varepsilon > 0 \text{ and } (\alpha-1)\varepsilon + \alpha &> 0 \\ \Leftarrow \varepsilon > 0 \text{ and } \varepsilon &< \frac{\alpha}{1-\alpha} \end{aligned} \tag{4.20}$$

  where $\alpha \in \mathbb{R}^+$ is the mass-proportionnal damping parameter. Finally, the function $V(x)$ is positive definite if $0 < \varepsilon < \frac{\alpha}{1-\alpha}$.

  Remark: *The upper bound comes from the case where $\alpha < 1$. In the other case ($\alpha > 1$), a sufficient condition is simply $\varepsilon > 0$. In this case, condition (4.16) becomes more conservative.*

- Then, $\dot{V}(x)$ must be negative definite; it writes:

$$\begin{aligned}\dot{V}(x) &= x^T\left((*) + \begin{pmatrix}(1+\varepsilon)\mathbf{M} & \varepsilon\mathbf{M} \\ \varepsilon\mathbf{M} & (1+\varepsilon)\mathbf{K}+\varepsilon\mathbf{D}\end{pmatrix}\begin{pmatrix}-\mathbf{M}^{-1}\mathbf{D} & -\mathbf{M}^{-1}\mathbf{K} \\ I & 0\end{pmatrix}\right)x \\ &= x^T\left((*) + \begin{pmatrix}-(1+\varepsilon)\mathbf{D}+\varepsilon\mathbf{M} & -(1+\varepsilon)\mathbf{K} \\ (1+\varepsilon)\mathbf{K} & -\varepsilon\mathbf{K}\end{pmatrix}\right)x \\ &= x^T\left(\begin{pmatrix}-2(1+\varepsilon)\mathbf{D}+2\varepsilon\mathbf{M} & 0 \\ 0 & -2\varepsilon\mathbf{K}\end{pmatrix}\right)x\end{aligned} \tag{4.21}$$

Again, $\varepsilon > 0 \Rightarrow -2\varepsilon\mathbf{K} < 0$. In addition, it holds:

$$\begin{aligned}&-2(1+\varepsilon)\mathbf{D}+2\varepsilon\mathbf{M} < 0 \\ \Leftrightarrow &-2\Big((\alpha+\varepsilon\alpha)\mathbf{M}+(\beta+\epsilon\beta)\mathbf{K}\Big)+2\varepsilon\mathbf{M} < 0\end{aligned} \tag{4.22}$$

If $\varepsilon > 0$, then $-2(\beta+\varepsilon\beta)\mathbf{K} < 0$. Finally, with $\varepsilon > 0$ a sufficient condition for $\dot{V}(x) < 0$ to hold is:

$$\begin{aligned}&-2(\alpha+\varepsilon\alpha)\mathbf{M}+2\varepsilon\mathbf{M} < 0 \\ \Leftrightarrow &-\alpha-\varepsilon\alpha+\varepsilon < 0 \\ \Leftrightarrow &\varepsilon < \frac{\alpha}{1-\alpha}\end{aligned} \tag{4.23}$$

Therefore, $\dot{V}(x) < 0$ holds if $0 < \varepsilon < \frac{\alpha}{1-\alpha}$.

■

The function $V(x)$ defined in (4.15) is thus a decreasing Lyapunov function for the open-loop system. Therefore, conditions (4.2) holds and the system is considered controllable, i.e. condition (4.3) holds too.

Finally, the following control law makes the origin of the large-scale system asymptotically stable (Thieffry et al. 2018):

$$u = -B^T P x = -B^T \begin{pmatrix}(1+\varepsilon)\mathbf{M} & \varepsilon\mathbf{M} \\ \varepsilon\mathbf{M} & (1+\varepsilon)\mathbf{K}+\varepsilon\mathbf{D}\end{pmatrix} x \tag{4.24}$$

4.4 Simulation results

These results are illustrated through simulation experiments. The robots chosen to validate the results is a two dimensional (2D) robots whose design is inspired from the robot presented in (Duriez 2013), shown in figure 4.1.

The 2D version of this robot is presented in figure 4.1. It is designed such that there exists only one equilibrium point at the rest position. The output is the position of one node located at the top center of the structure. The control law designed in equation (4.24) is applied to this example and results are shown in figure 4.2, that shows the trajectories of the end-effector in open and closed-loop. It is clear that the control law designed makes the robot converge faster to its final position.



Figure 4.1: *Soft robots whose design is described in (Duriez 2013). Real robot (top) and its 2D version (bottom).*
Left: rest position, Right: deformed position.

Figure 4.2: *Displacement of end effector in centimeters.*
Left: displacement along the horizontal axis, right the vertical axis.
Blue: open-loop trajectory; Red: closed-loop trajectory.

The design proposed presents a convenient way to derive a state feedback control. Obviously, it is impossible to measure the whole state of the robot. Therefore, the next step is to propose an output feedback control via a state observer. Due to the resulting high sized problem, this step is challenging especially considering the computational cost.

Part III

Model Order Reduction and Low-order Observer

Introduction

In part II, both continuous- and discrete-time large-scale state-space models have been presented. Based on these models and on the existence of an open-loop Lyapunov function, a large-scale controller has been designed. This first control design is however unusable in practice for state-feedback control due to the high number of variables in the state vector. To overcome this dimensionality issue, this part presents a model reduction algorithm in chapter 5 and the design of a low-order observer in chapter 6. Based on this low order model and observer, part IV presents different controller designs. Part of these results have been presented in (Thieffry et al. 2018, 2019).

Chapter 5 presents a reduction method to build a low order system based on which the design of both controller and observer is tractable. This reduction step includes a study of the reduction error required to design a robust low-dimension controller. In the literature, different reduction algorithms (moment matching, POD etc.) exist and a comparison of some of them is presented at the end of the chapter 5. The Proper Orthogonal Decomposition (POD) is well-suited for nonlinear model and for applications in simulation, as a lot of experiments need to be done but it requires an efficient simulation engine. In this work, for moment matching algorithm we use an available toolbox for MATLAB[2] (Poussot-Vassal & Vuillemin 2012) and for POD we use a plugin for SOFA dedicated to model order reduction[3] (Goury & Duriez 2018).

For soft robotics, placement of sensors is constrained by the robots geometry and behavior, e.g. the sensors should not bring additional stiffness and should permit to reconstruct the state vector. The number of sensors is also limited and thus, it is not possible to measure the whole shape of the robot. The state of the robot is therefore unknown in real-time. This manuscript deals with low-dimension state feedback controllers and thus requires the knowledge of the reduced order state. To use the controller in practice, chapter 6 proposes a reduced-order observer based on the low-order model described in chapter 5.

[2]MORE toolbox, https://w3.onera.fr/more/

[3]Model Order Reduction plugin for SOFA, https://project.inria.fr/modelorderreduction/

As for the linearization assumption, the reduction algorithms bring errors in the model that have to be studied in order to guarantee robustness of the closed-loop. To deal with these modeling errors, an unknown input observer is designed to reconstruct the low order state from the inputs and outputs of the robots. This low dimension state and the modeling errors will later be used in part IV to design a robust controller.

Chapter 5

Reduced Order Models

In soft robotics, model order reduction is used to achieve different objectives: to speed up the computation time of the simulation to reach real-time performances (Goury & Duriez 2018) or to simplify the control design. Model reduction algorithms have been widely studied and many theoretical results can be found in the literature as in (Antoulas 2005, Astolfi 2010, Benner et al. 2017).

This chapter presents model reduction algorithms used to overcome the dimensionality issues in order to design both reduced controller and observer.

Remark : For sake of clarity, the methods are presented for the continuous-time models but are later on applied to both the continuous and discrete-time systems.

Contents

5.1 Presentation of Reduction Algorithms 56
5.1.1 Proper Orthogonal Decomposition (POD) 57
5.1.2 Balanced Truncation 60
5.1.3 Approximation by moment matching 60
5.2 Low Order Model 61
5.3 Comparison of Reduction Methods 62
5.3.1 Proper Orthogonal Decomposition (POD) Applied to Soft Robots 63
5.3.2 Frequency Comparison: Bode plot 64
5.3.3 Conclusions on Reduction Methods 64

5.1 Presentation of Reduction Algorithms

In this work, we construct the low-order system using projection-based model reduction. Two main categories of algorithm are found in the literature: Singular Value Decomposition (SVD) based methods and Krylov (moment-matching) based methods. Balanced Truncation and Proper Orthogonal Decomposition (POD) are two SVD based methods, the first one is based on a dynamical model, the second one depends on simulation data. Balanced Truncation requires the computation of the system's gramian, which can sometimes be computationally expensive, but has the major advantage to offer an a priori error bound. The second method, POD, also offers such a bound and is in addition directly usable for non-linear large-scale systems. POD uses simulation data to perform the reduction, which requires an efficient simulation engine and the results strongly depend on the excitation signals used in the simulation to acquire the data.

For moment-matching reduction algorithms, one can find the Iterative Rational Krylov Algorithm (Gugercin et al. 2008), and its Multi-Input-Multi-Output (MIMO) version (Van Dooren et al. 2008). These methods are computationally tractable even for very large-scale systems. In this work, we can use any of the aforementioned reduction methods, for which a summarized theoretical description is given hereafter, for a detailed review see for instance (Vuillemin 2014, Benner et al. 2017).

Consider a nonlinear model:

$$\dot{x}(t) = f(x(t), u(t)) \ , \ x \in \mathbb{R}^n, \ n \gg 1 \tag{5.1}$$

projection-based model order reduction consists of decomposing the full-order state x into two parts, a low-order state $x_r \in \mathbb{R}^r$ and a neglected state $x_{\bar{r}} \in \mathbb{R}^{n-r}$ such that:

$$x = V_r x_r + V_{\bar{r}} x_{\bar{r}} \quad \text{with} \quad \begin{cases} x_r & = W_r^T x \\ x_{\bar{r}} & = W_{\bar{r}}^T x \end{cases} \tag{5.2}$$

The problem is thus to find two projectors $V_r \in \mathbb{R}^{n \times r}$ and $W_r \in \mathbb{R}^{n \times r}$ to compute a x_r with $r \ll n$ such that:

$$x_r = W_r^T x \quad ; \quad x \approx V_r x_r \tag{5.3}$$

In other words, the approximation method consists of finding the matrix V_r and W_r such that:

$$\dot{x}_r(t) - W_r^T f(V_r x_r(t), u(t)) = 0 \tag{5.4}$$

This provides an approximation of the large-scale system:

$$\dot{x}_r(t) = W_r^T f(V_r x_r(t), u(t)) \ , \ x_r \in \mathbb{R}^r, \ r \ll n \tag{5.5}$$

The problem consists in finding a computationnally efficient procedure so that the approximation error is small, in the sense of a given measure.

5.1.1 Proper Orthogonal Decomposition (POD)

The Proper Orthogonal Decomposition[1] method (POD) is a method which is widely used for nonlinear applications. The underlying idea is to collect samples of the state of the studied system and find a reduced basis that approximates these samples. This method is well suited for applications where simulation is available, as it is easy to get a collection of samples of states. The algorithm of POD can be summarized as: compute a singular value decomposition (SVD) of the snapshots and truncate depending on the decay rate of the singular values.

The first step of the POD is to compute the snapshot space. It can be generated in an offline, expensive, stage. During these simulations, we save the state at each time step. The time step has to be chosen small enough to capture the dynamical behaviour of the structure and the number of time-snapshots must be large enough to consider these simulations sufficiently exhaustive, so as to capture any situation the structure may encounter in the online stage.

Let $\Sigma : \dot{x} = f(x(t), u(t))$ be a nonlinear system and let S be a collection of $\mathcal{N}$ snapshots.

$$S = \begin{pmatrix} x(t_1) & x(t_2) & \dots & x(t_N) \end{pmatrix} \in \mathbb{R}^{n \times \mathcal{N}}$$

Computing a SVD of this matrix, we get Φ, the left singular matrix of S :

$$\Phi \Sigma \Omega^T = S \tag{5.6}$$

Let us define Φ_r as the r^{th} first columns of Φ, we have the following approximation:

$$S = \Phi \Sigma \Omega^T = \Phi_r \Sigma_r \Omega_r^T + \Delta \tag{5.7}$$

where Σ_r contains the r-first singular values of S and Δ represents the model order reduction errors.

Then, the projectors V_r and W_r are simply obtained as:

$$V_r = W_r = \Phi_r \tag{5.8}$$

[1] also known as Principal Component Analysis (PCA)

so that it holds:

$$x_r = W_r^T x \ ; \ x \approx V_r x_r \tag{5.9}$$

In addition, projectors are orthogonal and it holds:

$$W_r^T V_r = I_r \text{ and } W_r^T V_{\bar{r}} = 0 \tag{5.10}$$

This method offers the advantage of being easy to implement and provides the computation of the 'non-reduced' state $x_{\bar{r}}$ as $V_{\bar{r}}$ is directly obtained from the SVD decomposition, as it is made of the last $n-r$ columns of V. Coming back to equation (5.6):

$$\Phi \Sigma \Omega^T = S \tag{5.11}$$

Again, Φ_r are the r^{th} first columns of Φ, and $\Phi_{\bar{r}}$ are the $n-r$ last columns of Φ such as:

$$\Phi = \begin{pmatrix} \Phi_r & \Phi_{\bar{r}} \end{pmatrix}, \quad \Phi_r \in \mathbb{R}^{n \times r} \ , \quad \Phi_{\bar{r}} \in \mathbb{R}^{n \times (n-r)} \tag{5.12}$$

We thus have

$$S = U \Sigma \Omega^T = \begin{pmatrix} \Phi_r & \Phi_{\bar{r}} \end{pmatrix} \begin{pmatrix} \Sigma_r & \\ & \Sigma_{\bar{r}} \end{pmatrix} \begin{pmatrix} \Omega_r & \Omega_{\bar{r}} \end{pmatrix} \tag{5.13}$$

that leads to the exact relation between x, x_r and $x_{\bar{r}}$:

$$\begin{aligned} x_r = W_r^T x \ ; \ x_{\bar{r}} = W_{\bar{r}}^T x \\ x = V_r x_r + V_{\bar{r}} x_{\bar{r}} \end{aligned} \tag{5.14}$$

The POD algorithm has the advantage of being suitable for either non-linear or linear systems, without assumptions on the size of the original system. Moreover, only the continuous part has been presented above, but the method is applicable to discrete-time models. This method is easy to implement and provides an exact relationship between the full order model and the reduced order one.

The first drawback is that it requires a computationally expensive stage where the simulations are performed to save the snapshots. A second drawback is that as the method is based on simulation data, there is no guarantee for the situations not been taken into account in the offline stage.

For the pressurized robot, the high number of actuators yield complex shapes. Figure 5.1 shows some of the deformation of this robot.

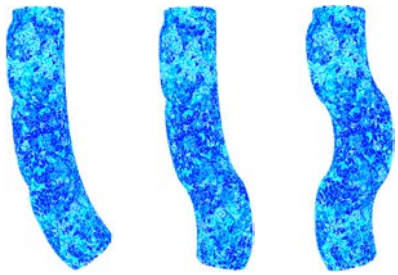


Figure 5.1: *Examples of deformations for the pressurized robot in one direction. This deformations may appear in any directions of space by coupling the actuators, yielding to a high number of possible deformations.*

Structure Preservation : Displacement and Velocity Snapshots

Note that different choices of snapshots can be made to perform the reduction; i.e. acceleration, velocity or position could be used as snapshots. From our initial testing, we found that using velocity and position was giving us more accurate results. At each time step, the matrices S_q and S_v are enriched with the value of the position and velocity vectors, q and $v = \dot{q}$ of the robot. We therefore obtain two snapshot matrices:

$$\begin{aligned} S_q &= \begin{pmatrix} q_{t0} & q_{t1} & \dots & q_{tf} \end{pmatrix} \\ S_v &= \begin{pmatrix} v_{t0} & v_{t1} & \dots & v_{tf} \end{pmatrix} \end{aligned} \tag{5.15}$$

Matrices V and W are then obtained performing a SVD of S_q and S_v. Depending on the decay rate of the singular values of Σ_p and Σ_v, a low-order approximation is obtained.

$$\begin{aligned} S_q &\approx \Phi_{q_r} \Sigma_{q_r} \Omega_{q_r}^T \\ S_v &\approx \Phi_{v_r} \Sigma_{v_r} \Omega_{v_r}^T \end{aligned} \tag{5.16}$$

Finally, the projectors V and W are defined as:

$$\begin{aligned} V_r = W_r &= \begin{pmatrix} \Phi_{v_r} & 0 \\ 0 & \Phi_{q_r} \end{pmatrix} \\ V_{\bar{r}} = W_{\bar{r}} &= \begin{pmatrix} \Phi_{v_{\bar{r}}} & 0 \\ 0 & \Phi_{p_{\bar{r}}} \end{pmatrix} \end{aligned} \tag{5.17}$$

so that Φ_p and Φ_v multiply respectively the position and velocity vectors q and v:

$$x_r = W_r^T x = \begin{pmatrix} \Phi_{v_r}^T v \\ \Phi_{q_r}^T q \end{pmatrix} = \begin{pmatrix} v_r \\ q_r \end{pmatrix} \tag{5.18}$$

5.1.2 Balanced Truncation

The balanced truncation consists of deleting states that are both hard to reach and hard to observe. Let consider a system H, it is said to be balanced if its controllability $\mathcal{P}$ and observability $\mathcal{O}$ gramians are diagonal and equals (Σ hereafter), such as:

$$\mathcal{P} = \mathcal{O} = \Sigma \tag{5.19}$$

Every reachable and observable system can be transformed to a balanced form through a change of basis. The matrix Σ is a diagonal matrix that can be written as $\mathbf{diag}(\Sigma_1, \Sigma_2)$ where Σ_1 gathers the values of interest and Σ_2 the values to be ignored.

$$\Sigma = \begin{pmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{pmatrix} \tag{5.20}$$

The reduced order system is obtained keeping only Σ_1.

The $\mathcal{H}_\infty$ norm of the error between the original large-scale system H and the reduced order one H_r is bounded by twice the sum of the neglected Hankel singular values (Antoulas 2005):

$$\|H - H_r\|_{\mathcal{H}_\infty} \leq 2 \sum_{i=r+1}^{n} \sigma_i \tag{5.21}$$

This method offers interesting properties, such has stability preservation and boundedness of the error between the original large scale model and the reduced order one. However, it may numerically fail for too high dimensions.

Balanced truncation was initially restricted to linear system but has been extended to nonlinear systems, for more details about balanced truncation applied to nonlinear systems see e.g. (Besselink et al. 2014, Lall et al. 2002). In addition it was initially designed for stable system, for which balanced truncation has the good property to preserve stability. Works presented in (Barrachina et al. 2005) or (Zhou et al. 1999) give ways to implement balanced truncation algorithms for unstable systems.

Remark: POD and Balanced Truncation algorithms share some properties and combining the best of each method yields better results (Rowley 2005).

5.1.3 Approximation by moment matching

This method is still a projection based method but it is no longer based on the input-output energy transfer. Model reduction by moment matching consists in finding a low order system that interpolates the initial large scale system at selected points in the complex place (Vuillemin 2014).

Moment matching problem: Given a n-th order model H which transfer function $H(s)$ is decomposed at $\sigma \in \mathbb{C}$ as:

$$H(s) = \sum_{i=0}^{\infty} \eta_i(\sigma) \frac{(s-\sigma)^i}{i!} \tag{5.22}$$

the moment matching problem consists in finding a reduced order model H_r which first r-moments $\hat{\eta}_i(\sigma)$ at σ satisfy

$$\hat{\eta}_i(\sigma) = \eta_i(\sigma) \ , \ i = 1, \dots, r \tag{5.23}$$

This problem is refered to with different names, depending on the point σ at which the moments have to be matched. At $\sigma = 0$, this problem is a Padé approximation, at $\sigma = \infty$, it is a partial realisation problem, otherwise, it is a rational interpolation problem.

A commonly used moment matching reduction algorithm is the Iterative Rational Krylov Algorithm (IRKA), detailed in (Gugercin et al. 2008).

Remark: If only POD is directly adapted for nonlinear systems, IRKA algorithm has also been extended to the bilinear case (Benner & Breiten 2012, Benner et al. 2011).

5.2 Low Order Model

The large scale state x is thus decomposed into a low-order state $x_r \in \mathbb{R}^r$ and a neglected state $x_{\bar{r}} \in \mathbb{R}^{n-r}$ such that:

$$x = V_r x_r + V_{\bar{r}} x_{\bar{r}} \quad \text{with} \quad \begin{cases} x_r &= W_r^T x \\ x_{\bar{r}} &= W_{\bar{r}}^T x \end{cases} \tag{5.24}$$

The projectors W and V are orthogonal, it holds $W_r^T V_r = I$ and $W_r^T V_{\bar{r}} = 0$. Consider now a linear model, full order model (FOM):

$$(FOM) \ : \ \begin{cases} \dot{x} = Ax + Bu \\ y = Cx \end{cases} \tag{5.25}$$

The exact dynamics of the low order state writes:

$$(EROM) \ : \ \begin{cases} \dot{x}_r = W_r^T A V_r x_r + W_r^T B u + W_r^T A V_{\bar{r}} x_{\bar{r}} \\ \dot{x}_{\bar{r}} = W_{\bar{r}}^T A V_{\bar{r}} x_{\bar{r}} + W_{\bar{r}}^T B u + W_{\bar{r}}^T A V_r x_r \\ y_r = C V_r x_r + C V_{\bar{r}} x_{\bar{r}} \end{cases} \tag{5.26}$$

If the model reduction step is accurate enough, one can consider the following Approximated Reduced Order Model (AROM) for the control and observation problem:

$$(AROM) \ : \ \begin{cases} \dot{x}_r = W_r^T A V_r x_r + W_r^T B u = A_r x_r + B_r u \\ y_r = C V_r x_r = C_r x_r \end{cases} \tag{5.27}$$

However, in some cases, a too sharp model reduction will introduce reduction errors that have a non neglictible effect on the model.

The controller designed based on this low order model should ensure stability of the exact reduced order model (EROM) (5.26), therefore the states $x_{\bar{r}}$ have to be taken into account. The dynamics of the low order state x_r only depends on reduced order matrices A_r, B_r, C_r, based on which the design of controllers and observers are computationaly tractable.

However, a controller designed for the low order model (5.27) can lead to unstable closed-loop system when applied to the large-scale one. To avoid this, one aim at designing a robust low dimensions controller. This can be done by studying the exact dynamics of the low order state in (5.26). However, it depends on the matrix $W_r^T A V_{\bar{r}} \in \mathbb{R}^{r \times (n-r)}$ which is of large dimensions. This does not help to reduce the complexity for the controller design.

Considering that the term based on the unknown state $x_{\bar{r}}$ should be small compared to x_r (due to the model reduction), a way to tackle this issue is to consider the error reduction as an unknown disturbance d_u. Re-writing this model in a more efficient manner, involving reduced dimensions matrices is done by decomposing the reduction error $W_r^T A V_{\bar{r}} x_{\bar{r}}$ into two orthogonal parts: an input disturbance d_u plus an unavoidable modeling error ε:

$$W_r^T A V_{\bar{r}} x_{\bar{r}} = B_r d_u + B_r^{\perp} \varepsilon = \begin{pmatrix} B_r & B_r^{\perp} \end{pmatrix} \begin{pmatrix} d_u \\ \varepsilon \end{pmatrix} \tag{5.28}$$

where $B_r^{\perp}$ is an orthogonal complement of B_r, such as $B_r^{\perp^T} B_r = 0$, and $B_r^{\perp^T} B_r^{\perp} = I$ and the matrix $\begin{pmatrix} B_r & B_r^{\perp} \end{pmatrix}$ has full rank.

Finally, with notations presented above and from (5.26) and (5.28), the low order dynamics writes:

$$M_r \ : \ \begin{cases} \dot{x}_r = A_r x_r + B_r u + B_r d_u + B_r^{\perp} \varepsilon \\ y_r = C_r x_r \end{cases} \tag{5.29}$$

5.3 Comparison of Reduction Methods

Balanced truncation offers an a priori error bound between the reduced order system and the large-scale one. This is often considered as the best reduction

method for control applications, but it has been pointed out that it could numerically fail when the order of the system to reduce is too high. We present hereafter a comparison between the results obtained with POD and ITIA algorithms.

5.3.1 Proper Orthogonal Decomposition (POD) Applied to Soft Robots

Singular Values of the Snapshot

We study here the singular value decomposition of the snapshot of two different robots using POD algorithm. The number of snapshots stored depends on the number of actuators, the objective is to test a maximum of possible combinations of actuation, so as to be as exhaustive as possible. For the cable-driven soft robot with 4 actuators there are 16 combination of actuation, for the pressurized robot with 12 acutators there are 4096 possible combinations.

The accuracy of the reduced model depends on the decay rate of the singular values of the snapshots. Figure 5.2 shows the evolution of the singular values of the position snapshots of the cable driven robot. The inflection point appears at the 67^{th} value. One can see a fast decay rate in the singular values plots for this model that is confirmed by the additional information given in the following table 5.1.

Remark: *Similar results are obtained for the velocity snapshots, where the inflection point appears at the* 58^{th} *value and where the three first singular values represent* 94.31% *of the total of the values.*

Figure 5.2: *Evolution of the singular values of the position snapshots of the cable driven soft robot.*
Left : singular values, Right: 10 *first singular values*

Figure 5.3 and table 5.2 show that few singular values are needed to capture the main behavior of the cable-driven soft robot.

Table 5.1: *Singular Values (SV) of the position snapshots of the cable-driven robot:*

	first singular value (SV)	3 first SV	67 first SV
% of the SV	79.12%	93.81%	99.99%

However, different results are obtained for the pressurized robots depicted in figure 3.6. The inflection point appears at the 88^{th} value, and then rapidly reaches the minimum. This is explained by the independance of the deformation modes of the structure. The first five represent 72.68% of the singular values of Σ.

Figure 5.3: *Evolution of the singular values of the position snapshots of the pressurized soft robot.*
Left : singular values, Right: 10 *first singular values*

Table 5.2: *Singular Values (SV) of the position snapshots of the pressurized robot:*

	first singular value (SV)	5 first SV	88 first SV
% of the SV	15.17%	72.68%	99.99%

5.3.2 Frequency Comparison: Bode plot

Let us consider the bode plot of the transfer between the first input and the first output of the model of the cable-driven robot presented in figure 3.1. Figure 5.4 presents the comparison between the frequency response of the large and reduced system using different algorithms: POD and ITIA. It shows that for very low order systems, ITIA algorithm is more accurate than POD when comparing the frequency response.

5.3.3 Conclusions on Reduction Methods

This sections show that POD and ITIA are two efficient algorithms to perform model order reduction for soft robotics applications. POD algorithm is well suited when a simulated model of the robot is available. When the number of

Figure 5.4: *Bode plot of the large model (blue) and reduced model (red) of the cable-driven robot.*
We consider only the first input and the first output of the system.
Top: reduced (red) and large (blue) system using POD
Left : order 4, right : order 8
Bottom : large (blue) and reduced (red) model using ITIA.
Left : order 4, right : order 8

actuators increase, so does the complexity of the method and complex shapes yield large reduction errors.

For linear systems; tangential interpolation gives accurate results and there exists off-the-shell algorithms implementing this method (Poussot-Vassal & Vuillemin 2012).

Chapter 6

Low-order Observer

To use the state feedback controller in practice, where neither x nor x_r are available from sensors, an observer is designed to reconstruct x_r from y and u. The reduction error is taken into account in the computation to design an observer that is robust according to the reduction error (Thieffry et al. 2019). To do so, an unknown input observer, in the form of a PI-observer as in (Ichalal et al. 2009), reconstructs x_r and considers the reduction error as a disturbance, i.e. the unknown input.

6.1 Introduction

Considering the low-order model without reduction error:

$$\begin{cases} x_{r_+} = A_r x_r + B_r u \\ y_r = C_r x_r \end{cases} \tag{6.1}$$

A Luenberger-observer for this system would be:

$$\begin{cases} x_{r_+} = A_r \hat{x}_r + B_r u + K_o(y_r - \hat{y}_r) \\ \hat{y}_r = C_r \hat{x}_r \end{cases} \tag{6.2}$$

Using classical tools of automatic control, the computation of the observer gain K_o is easily solved. However, the system to control is not the low order model without reduction error, but the large-scale one. As the design of an observer

for the large-scale model is computationnaly untractable, a good compromise is to design an observer for the low-order system taking into account reduction error. This is done by designing an unknown input observer.

6.2 Discrete-time Unkown Input Observer Design

From equation 5.29 the discrete-time low order model writes:

$$\begin{cases} x_{r_+} = A_r x_r + B_r u + B_r d_u + B_r^{\perp} \varepsilon \\ y_r = C_r x_r \end{cases} \tag{6.3}$$

where d_u is the unknown input of the system and ε gathers the remaining modeling errors.

The design is based on a so-called unknown input PI-observer. It assumes that the dynamics of d_u can be captured via a cascade of integrators, i.e. the p^{th} variation of d_u is zero. Under this assumption, denoting $\tilde{\mathbf{B}}_r = \begin{pmatrix} B_r & 0 & \dots & 0 \end{pmatrix}$ and defining an extended state $(\hat{x}_r^T \ \hat{D}_u^T)^T$, the observer writes:

$$\begin{pmatrix} \hat{x}_{r_+} \\ \hat{D}_{u_+} \end{pmatrix} = \begin{pmatrix} A_r & \tilde{\mathbf{B}}_r \\ 0 & J \end{pmatrix} \begin{pmatrix} \hat{x}_r \\ \hat{D}_u \end{pmatrix} - \mathbf{K_o} C_r (x_r - \hat{x}_r) + \begin{pmatrix} B_r \\ 0 \end{pmatrix} u \tag{6.4}$$

with

$$\underbrace{\begin{pmatrix} \hat{d}_{u_+} \\ \hat{d}^{(2)}_{u_+} \\ \vdots \\ \hat{d}^{(p-1)}_{u_+} \\ \hat{d}^{(p)}_{u_+} \end{pmatrix}}_{\hat{D}_{u_+}} = \underbrace{\begin{pmatrix} I & I & 0 & \dots & 0 \\ 0 & I & I & \ddots & 0 \\ \vdots & 0 & \ddots & \ddots & 0 \\ \vdots & \vdots & \ddots & I & I \\ 0 & 0 & \dots & 0 & I \end{pmatrix}}_{J} \underbrace{\begin{pmatrix} \hat{d}_u \\ \hat{d}^{(2)}_u \\ \vdots \\ \hat{d}^{(p-1)}_u \\ \hat{d}^{(p)}_u \end{pmatrix}}_{\hat{D}_u} \tag{6.5}$$

To keep the observability property, the following constraint must be satisfied (Ichalal et al. 2009):

$$rank(B_r) \leq p \tag{6.6}$$

where p is the number of outputs. Let us define the observation error e_o:

$$\begin{aligned} e_o &= \begin{pmatrix} x_r - \hat{x}_r \\ D_u - \hat{D}_u \end{pmatrix} = \begin{pmatrix} e_{o_x} \\ e_{o_d} \end{pmatrix} \\ \Rightarrow e_{o_+} &= \underbrace{\left[\begin{pmatrix} A_r & \tilde{\mathbf{B}}_r \\ 0 & J \end{pmatrix} - \mathbf{K_o} (C_r \ O) \right]}_{\mathbf{A_o - K_o C_o}} e_o + \begin{pmatrix} B_r^{\perp} \\ 0 \end{pmatrix} \varepsilon \end{aligned} \tag{6.7}$$

The computation of the matrix $\mathbf{K_o}$ is detailed in section 6.3.

6.3 Observer design according to modeling error issue

Let us recall the definition of *Input to State Stability (ISS)*:

Theorem 5 of (Sontag 2008):

A system with state x and input u is ISS if and only if it admits a smooth Lyapunov function $V(x)$ such that its variation satisfies:

$$\Delta V(x,u) \leq -\alpha(\|x\|) + \gamma(\|u\|) \tag{6.8}$$

where $\alpha, \gamma \in \mathcal{K}_\infty$. □

In the remainder of this document, we will use quadratic $\mathcal{K}_\infty$ functions defined as:

$$\alpha(\|x\|) = \alpha x^T M x, \quad \alpha > 0, \ M > 0 \tag{6.9}$$

The ISS property will come at hand to study the stability property of the closed-loop according both to the modeling error ϵ and the input disturbance d_u.

The first objective is to compute $\mathbf{K_o}$ in (6.7) while minimizing the impact of the unknown modeling error ε on the observer error e_o that writes:

$$\begin{aligned}
&\text{Find } \mathbf{K_o} \text{ and } P_o \text{ in } V_o(e_o) = e_o^T P_o e_o \text{ and minimize } \gamma_o \text{ for the model:} \\
&e_{o_+} = (\mathbf{A_o} - \mathbf{K_o C_o}) e_o + \begin{pmatrix} B_r^\perp \\ 0 \end{pmatrix} \varepsilon \\
&\text{such that:} \\
&\Delta V_o < -\alpha_o e_o^T P_o e_o + \gamma_o \varepsilon^T \varepsilon \\
&\text{with : } \alpha_o > 0, \ \gamma_o > 0, \ P_o > 0
\end{aligned} \tag{6.10}$$

(6.10) is equivalent to:

$$\begin{aligned}
&\left[* \right] P_o \left[(\mathbf{A_o} - \mathbf{K_o C_o}) e_o + \begin{pmatrix} B_r^\perp \\ 0 \end{pmatrix} \varepsilon \right] - e_o P_o e_o && < -\alpha_o e_o^T P_o e_o \\
& && \quad + \gamma_o \varepsilon^T \varepsilon \\
\Leftrightarrow &\begin{pmatrix} e_o^T & \varepsilon^T \end{pmatrix} \left[* \right] P_o \begin{bmatrix} \mathbf{A_o} - \mathbf{K_o C_o} & \begin{pmatrix} B_r^\perp \\ 0 \end{pmatrix} \end{bmatrix} \begin{pmatrix} e_o \\ \varepsilon \end{pmatrix} & & \\
&+ \begin{pmatrix} e_o^T & \varepsilon^T \end{pmatrix} \begin{pmatrix} -P_o + \alpha_o P_o & 0 \\ 0 & -\gamma_o I \end{pmatrix} \begin{pmatrix} e_o \\ \varepsilon \end{pmatrix} && < 0
\end{aligned} \tag{6.11}$$

Using Schur's complement, this is equivalent to the following matrix inequality condition:

$$(6.10) \Leftrightarrow \begin{pmatrix} -P_o + \alpha_o P_o & 0 & * \\ 0 & -\gamma_o I & * \\ P_o \mathbf{A_o} - \tilde{\mathbf{K}}_o \mathbf{C_o} & P_o \begin{bmatrix} B_r^{\perp} \\ 0 \end{bmatrix} & -P_o \end{pmatrix} < 0 \tag{6.12}$$

with the classical change of variables $\tilde{\mathbf{K}}_o = P_o \mathbf{K_o}$. It ensures that for large enough time, the trajectory error converges to a hyperball whose radius depends on γ_o and α_o. Therefore the observer error converges into the region defined as:

$$e_o P_o e_o \leq \frac{\gamma_o \epsilon^T \epsilon}{\alpha_o} \tag{6.13}$$

6.4 Summary on Observer Design

The model reduction algorithm presented in chapter 5 enables the design of a low-order observer. To guarantee the robustness of the reduced state reconstruction, model reduction errors are taken into account in the design of the observer via Input-To-State Stability condition. This observer makes it possible to apply a low-dimension state feedback control, whose design is detailed in part IV.

Part IV

Low-dimensions Model-based Controller

Introduction

Commonly used controllers for linear system design are the output feedback controllers based on a state feedback reconstructed approach; this part is entirely dedicated to this kind of control laws.

Part II presented large-scale state-space models and a large-scale controller based on the existence of an open-loop Lyapunov function. Part III presented the model reduction algorithms and the design of a low order observer. This part presents new methods to build low-dimension controllers with a stability constraint for the full order model. This is achieved with two differents strategies, detailed in chapters 7 and 8.

It is straightforward to design a controller based on the low-order model without considering the large-scale plant. For a system with a reasonable number of states, classical tools of automatic control (lqr, pole placement etc.) give desired results. However, this contoller is meant to be used on a real robot which exhibits high order dynamics that were neglected during the design phase. This may lead to poor-performance or unstable closed loop. Properties of the reduced closed-loop system should also be preserved when studying the large-scale closed-loop.

As an example, let us consider the discrete-time model of the cable-driven soft robot:

$$x_+ = Ax + Bu \ ; \ x_{r_+} = A_r x_r + B_r u \tag{6.14}$$

The eigenvalues of the full-order model are in:

$$\begin{aligned} real - part &: [0.35, 0.78] \\ im.part &: [-0.28, 0.28] \\ abs.value &: [0.35, 0.83] \end{aligned} \tag{6.15}$$

The low-order model is obtained using moment-matching method and its

eigenvalues p are:

$$p = \begin{pmatrix} 0.4e-2 \\ 0.59 \\ 0.65+0.31i \\ 0.65-0.31i \\ 0.71+0.35i \\ 0.71-0.35i \end{pmatrix} \tag{6.16}$$

One may want to cancel the oscillations of the system using low-dimension state-feedback $u = -Fx_r$ with pole placement method to set the poles of the closed-loop as a desired real location p_d:

$$p_d = abs(p) \tag{6.17}$$

This control law leads to a low-order closed-loop model $x_{r_+} = (A_r - B_r F)x_r$, where the poles of $A_r - B_r F$ equals p_d.

However, let us apply this control low to the full-order model, yielding to the corresponding closed-loop $x = (A - BFW_r^T)x$. The stability and performances of the closed-loop system are given by the eigenvalues of $A - BFW_r^T$, denoted p_{cl} that are in:

$$\begin{aligned} real-part &: [0.16, 1.14] \\ im.part &: [-0.22, 0.22] \\ abs.value &: [0.16, 1.17] \end{aligned} \tag{6.18}$$

Even if the low-order closed-loop model has desired properties, the full-order model exhibits unwanted behavior in closed-loop.

Our objective is also to design a low-dimension controller while guaranteeing properties for the large-scale closed-loop system. This problem has already been studied in (Benner et al. 2018), where the authors propose a low dimension controller with a stability constraint on the full order model using $\mathcal{H}_\infty$ controllers. In this manuscript, we aim at designing low dimension controller with a stability constraint for the full order model using Linear Matrix Inequalities (LMI) constraints problems.

Chapter 7 uses the knowledge of an open-loop Lyapunov function to study the large-scale stability. The design of a low-order observer is done that permits to validate the method on a physical setup. The control design is a low-dimension state feedback controller with a stability constraint on the full order model, expressed as a LMI constraint (Thieffry et al. 2019, Katzschmann, Thieffry et al. 2019).

To extend the workspace of the robot, chapter 8 proposes a trajectory tracking controller. We study the reduction error and use Input-to-State (ISS) stability

properties to study and prove the stability of the large-scale closed-loop system[1]. Experimental validations of the approach and a study of the convergence region of the closed-loop are conducted.

[1]These results have been accepted for publication: *Trajectory Tracking Control Design for Large Scale Linear Dynamical Systems with applications to Soft Robotics, M. Thieffry, A. Kruszewski, T.M. Guerra, C. Duriez, in IEEE Transactions on Control Systems Technology*

Chapter 7

Low-dimension Controller with Large-Scale Stability using Energy-based Lyapunov Functions

Starting from a large-scale LTI system and using one of the reduction algorithm described in chapter 5, a reduced order LTI system of reasonable size is obtained and classical tools from control theory come at hand to design a feedback controller. This method has the advantage to be simple to implement.

However, due to the approximation errors, controllers designed for this low order model can lead to unstable closed-loop systems when applied to the large-scale plants. This chapter aims at providing new solutions to design controller based on low-dimension models while guaranteeing properties regarding the full order model.

Contents

7.1 Closed-loop large-scale Lyapunov function 78

7.2 Simulation Experiments 79

7.2.1 Cable-driven trunk-like robot 79

7.2.2 Pressurized 3D arm 79

7.3 Real-time Experiments 82

7.1 Closed-loop large-scale Lyapunov function

Let us consider a natural (open loop) stable equilibrium of a robot. In this case, energy-like functions are possible Lyapunov candidate functions.

This choice of Lyapunov function reduces the complexity in the computation of the large-scale Lyapunov function, that would imply thousands of variables. However, in the case of a state feedback this does not help to compute the controller gains because of the number of decision variables. The idea is to use both Lyapunov stability and the model reduction to design a low dimension state feedback controller, while proving the stability of the original full order system using a large-scale Lyapunov function.

The proposed controller is a low dimension state feedback controller:

$$u = -Lx_r \tag{7.1}$$

The algorithm to obtain the reduced state x_r is presented in chapter 5, that also provides the dynamic of the low order state:

$$\dot{x}_r = A_r x_r + B_r u + A_{\bar{r}} x_{\bar{r}} \tag{7.2}$$

where $x_r = W_r^T x$ and $x = V_r x_r + V_{\bar{r}} x_{\bar{r}}$, V and W being two orthogonal projectors.

Let us define the matrix $P > 0$ by recalling the Lyapunov function $V(x)$, described in (4.15):

$$V(x) = x^T P x = x^T \begin{pmatrix} (1+\varepsilon)\mathbf{M} & \varepsilon\mathbf{M} \\ \varepsilon\mathbf{M} & (1+\varepsilon)\mathbf{K} + \varepsilon\mathbf{D} \end{pmatrix} x \tag{7.3}$$

where ε is a parameter added to reduce conservatism, when it is set to zero $V(x)$ is the energy of the system.

The derivative of the function $V(x)$ along the trajectories of the closed-loop writes:

$$\dot{V}(x) = x^T \Big((*) + P(A - BLW_r^T) \Big) x \tag{7.4}$$

This defines a non-convex problem. One way to write it as a convex problem is to write $\dot{V}(x)$ with respect to the projected states x_r and the $x_{\bar{r}}$:

$$\begin{aligned} \dot{V}(x) &= \begin{pmatrix} x_r \\ x_{\bar{r}} \end{pmatrix}^T \begin{pmatrix} V_r^T \\ V_{\bar{r}}^T \end{pmatrix} \Big((*) + P(A - BLW_r^T) \Big) \begin{pmatrix} V_r & V_{\bar{r}} \end{pmatrix} \begin{pmatrix} x_r \\ x_{\bar{r}} \end{pmatrix} \\ \dot{V}(x) &= \begin{pmatrix} * \end{pmatrix} + \begin{pmatrix} x_r \\ x_{\bar{r}} \end{pmatrix}^T \begin{pmatrix} V_r^T PAV_r - V_r^T PBL & V_r^T PAV_{\bar{r}} \\ V_{\bar{r}}^T PAV_r - V_{\bar{r}}^T PBL & V_{\bar{r}}^T PAV_{\bar{r}} \end{pmatrix} \begin{pmatrix} x_r \\ x_{\bar{r}} \end{pmatrix} \end{aligned} \tag{7.5}$$

This last equation (7.5) defines a LMI problem with respect to the matrix L. (The matrix P is defined in equation (7.3) and is not a decision variable.) The number of decision variables is low as the matrix $L \in \mathbb{R}^{m \times r}$ is of reduced dimensions. However, the number of constraints is still high (n), to guarantee the stability of the full order model.

The computation of the control gain L is made through a LMI problem[1] while optimizing the decay rate λ of the Lyapunov function $V(x)$:

$$\begin{gathered} \dot{V}(x) < \lambda V(x) \\ \Leftrightarrow \\ \begin{pmatrix} * \end{pmatrix} + \begin{pmatrix} V_r^T P A V_r - V_r^T P B L & V_r^T P A V_{\bar{r}} \\ V_{\bar{r}}^T P A V_r - V_{\bar{r}}^T P B L & V_{\bar{r}}^T P A V_{\bar{r}} \end{pmatrix} < \lambda \begin{pmatrix} (1+\varepsilon)\mathbf{M} & \varepsilon\mathbf{M} \\ \varepsilon\mathbf{M} & (1+\varepsilon)\mathbf{K} + \varepsilon\mathbf{D} \end{pmatrix} \end{gathered} \tag{7.6}$$

7.2 Simulation Experiments

7.2.1 Cable-driven trunk-like robot

This control design has first been proposed in (Thieffry et al. 2018) where only simulation were shown, as the design of the observer was not conducted yet. Simulation experiments are conducted on the cable driven robot presented on figures 3.1 and 3.2. For the reduction step, POD algorithm is used to obtain a low order system of dimension 6. Simulations are performed on the nonlinear full-order model.

First, the position and velocity vectors in open and closed-loop are compared in 7.1 and 7.2. It shows a clear diminution of oscillations while converging to the equilibrium point.

Then, the low order state in open and closed-loop are compared in 7.3.

Remark: *The resolution of the LMI* (7.6) *with* 48 *variables and with size* 9342×9342 *took* 75 *minutes on a Intel Core i7 CPU.*

7.2.2 Pressurized 3D arm

A first test is performed where the objective is to drive the simulated soft arm from a deformed shape to its rest position. The comparison of the position of the end-effector for this experiment in open and closed-loop is shown in Figure 7.4.

[1] solved with YALMIP for MATLAB (Löfberg 2004)

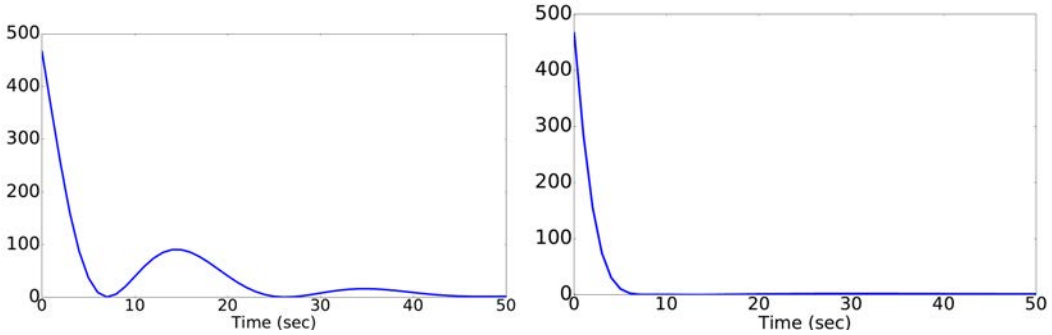


Figure 7.1: *Norm of the displacement vector $d = q - q_0$ (cm) in open-loop (left) and closed-loop (right) simulations.*

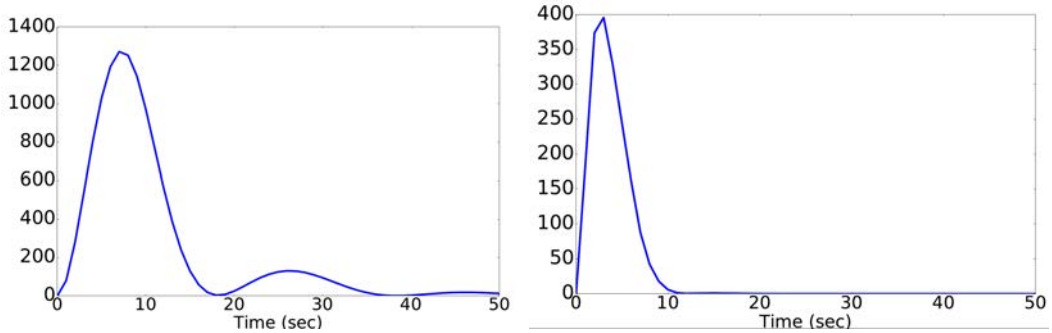


Figure 7.2: *Norm of the velocity vector $v = \dot{q}$ (cm/s^2) in open-loop (left) and closed-loop (right) simulations.*

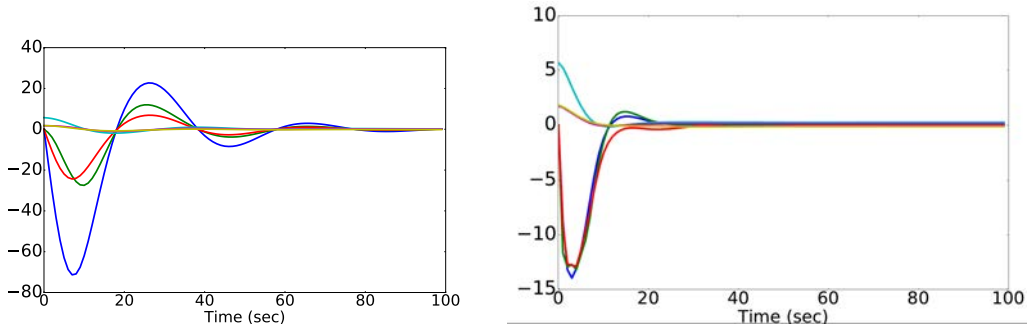


Figure 7.3: *Reduced order state in open-loop (left) and closed-loop (right) simulations.*

A second set of experiments is conducted where the arm starts from its rest shape position and converges to a deformed position; results are shown in Figure 7.5.

As one metric for evaluation of the results, we use the integral time absolute

Figure 7.4: *Simulation of closed-loop control, showing the motion of the end-effector going from an actuated state to the straight resting position. Red, blue and yellow lines show position of the end-effector along x, y and z axis.*

Figure 7.5: *Simulation of closed-loop control, showing the motion of the end-effector going from the straight resting position to an actuated state. Red, blue and yellow lines show position along x, y and z axis.*

error (ITAE) criterion defined as:

$$ITAE = \int_0^T t|e(t)|dt \tag{7.7}$$

This criterion is used to measure the overshoots and oscillations of the system response. To compare the controller performances, the value of the ITAE in closed-loop is compared with the one of the open-loop. In addition, the 3% settling time is given for the same experiments in open and closed loop; results are shown in Table 7.1. Results show a maximum gain of 67.53% in ITAE and 58.65% for the settling time. It also shows that the improvements are higher when the target is the position where the model of the robot has been linearized.

Table 7.1: *Control in Simulation: Comparison of Integral Time Absolute Error (ITAE) and 3% settling time in open and closed-loop.*

	to rest position	to curled position
ITAE in:		
open-loop	691.52	242.19
closed-loop	224.56	209.24
difference in %	67.53 %	13.61 %
3% settling-time in:		
open-loop	6.53 s	3.52 s
closed-loop	2.7 s	1.6 s
difference in %	58.65 %	54.55 %

7.3 Real-time Experiments

Thanks to the continuous time low order observer whose design is detailed in section 6.2, real-time experiments are conducted on the pressurized soft robot presented on figures 3.6 and 3.7. These results have been partly presented in (Thieffry et al. 2019, Katzschmann, Thieffry et al. 2019).

A closed loop controlled pose-to-pose motion starting from an actuated state and going to the straight resting pose is compared to the open-loop controlled scenario and results are shown in figure 7.6. The ITAE and 3% settling time

Figure 7.6: *Real-time closed-loop experiment, measuring the end-effector position. Red, blue and yellow lines show the position along the x, y and z axis.*

are gathered in Table 7.2 to compare the open-loop against the closed-loop experiment. The results show an improvement of 57% in the ITAE metric and an improvement of 49.62% in the 3% settling time metric. The difference between the simulated and experimental results are visible; the improvements in the experiments are lower than the ones found in simulation. This difference may come from different causes: the linearization assumption or from the actuators dynamics that are not modeled. Despite these difference between

simulation and reality, experimental results still show that the controller clearly improves the performance of the system.

Table 7.2: *Open and closed-loop experiments, comparing Integral Time Absolute Error (ITAE) and 3% settling time.*

	to rest position	to curled position
ITAE in:		
open-loop	691.52	242.19
closed-loop	297.33	218.95
difference in %	57 %	9.6 %
3% settling time in:		
open-loop	6.53 s	3.52 s
closed-loop	3.29 s	2.21 s
difference in %	49.62 %	37.22 %

CHAPTER 8

Trajectory Tracking for Large-Scale Linear Systems

The controller designed should ensure stability of the exact low order model (EROM) (5.26), the states $x_{\bar{r}}$ have to be taken into account. Considering that the term based on the unknown state $x_{\bar{r}}$ should be negligible compared to x_r (due to the model reduction), a way to tackle this issue is to consider the error reduction as an unknown disturbance d_u associated to an unavoidable modeling error ε. Thus, an adequate controller has to cope with these uncertainties to make them vanish via the robustness property of the feedback control law.

Contents

8.1 Introduction 86
8.2 Control Design 87
8.2.1 Reference Model 87
8.2.2 Observer-based output feedback 88
8.3 Computation of Controller Gains 90
8.3.1 Proof of stability for complete low order system 92
8.3.2 Summary of this closed-loop algorithm 93
8.4 Experimental Validation 94
8.4.1 Experimental setup 95
8.4.2 Study of Reduced Order Model 95
8.4.3 Validation of control design 97

8.1 Introduction

Let us start from a linear large-scale system given in equation (8.1).

$$M \ : \ \begin{cases} x_+ = Ax + Bu \\ y = Cx \end{cases} \tag{8.1}$$

where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, m being the number of actuators and $C \in \mathbb{R}^{p\times n}$ is a matrix defining the system outputs and p is the number of outputs.

For this model to be precise, the number of variables in the state vector x has to be significantly high and system (8.1) is thus a large-scale system with $n \gg 1$.

The large dimension of system (8.1) makes difficult the use of standard tools of automatic control (such as pole placement or LMI constraints problem) to design a controller for this system. To tackle this problem, we propose to use model order reduction methods to obtain a low order system that represents the full order system. Based on this low order state, the designs of both controller and observer are tractable.

The dynamics of the low order state writes:

$$\begin{cases} x_{r_+} = \underbrace{W_r^T A V_r}_{A_r} x_r + \underbrace{W_r^T B}_{B_r} u + W_r^T A V_{\bar{r}} x_{\bar{r}} \\ y_r = \underbrace{C V_r}_{C_r} x_r + C V_{\bar{r}} x_{\bar{r}} \end{cases} \tag{8.2}$$

The objective is to control the full order model with a low-dimension controller. Two possibilites arise: the first consists of completely neglect the influence of $x_{\bar{r}}$, the second is two study this reduction error in the design of the control law. Thus, an adequate controller has to cope with these uncertainties to make them vanish via the robustness property of the feedback control law.

From chapter 5 and equation (5.29), this model writes as:

$$M_r \ : \ \begin{cases} x_{r_+} = A_r x_r + B_r u + B_r d_u + B_r^{\perp} \varepsilon \\ y_r = C_r x_r \end{cases} \tag{8.3}$$

Lemma 2:

Assuming that $x_{\bar{r}}$ is bounded in a domain around the trajectory, then a bound of ε is given by (8.4):

$$\|\varepsilon\| \leq \eta \tag{8.4}$$

and an overestimated upper bound η of the modeling error ε is:

$$\begin{aligned} W_r^T A V_{\bar{r}} x_{\bar{r}} &= B_r d_u + B_r^{\perp} \varepsilon \\ B_r^{\perp T} W_r^T A V_{\bar{r}} x_{\bar{r}} &= B_r^{\perp T} B_r d_u + B_r^{\perp T} B_r^{\perp} \varepsilon \\ B_r^{\perp T} W_r^T A V_{\bar{r}} x_{\bar{r}} &= \varepsilon \\ \Rightarrow \|\varepsilon\| &\leq \|B_r^{\perp T} W_r^T A V_{\bar{r}} x_{\bar{r}}\| = \eta \end{aligned} \tag{8.5}$$

The study of the reduction error $W_r^T A V_{\bar{r}} x_{\bar{r}}$ gives a value of this bound. Not all reduction methods listed above provide the computation of the large matrix $V_{\bar{r}}$, as it is computationally expensive, but one can study this bound using:

$$\begin{aligned} x_+ &= Ax + Bu \\ W_r^T x_+ &= W_r^T A x + W_r^T B u \\ W_r^T x_+ &= W_r^T A (V_r x_r + V_{\bar{r}} x_{\bar{r}}) + W_r^T B u \\ W_r^T x_+ &= W_r^T A V_r W_r^T x + W_r^T A V_{\bar{r}} x_{\bar{r}} + W_r^T B u \\ \Rightarrow W_r^T A V_{\bar{r}} x_{\bar{r}} &= W_r^T x_+ - A_r W_r^T x - B_r u \end{aligned} \tag{8.6}$$

It is always possible to get an estimation of the upper bound of ε. For a simple trajectory, running simulations around the trajectory is sufficient. For a global bound, simulations can be provided covering the entire workspace of the effector and considering the worst case. Experimental values of this bound are provided in section 8.4, with the experimental validation of the method.

Hereafter, in this chapter, we present a trajectory tracking controller based on a low order model with uncertainties (see model M_r, equation (5.29)). The control strategy is presented in section 8.2 and 8.3, it includes both feed-forward and observer-based output feedback elements. Under the assumption made, the method proposed ensures the stability of the full order model M, equation (8.1), through linear matrix inequalities problems. The algorithm, computed thanks to the low-order model, is then applied to the full order model and real-time experiments are gathered in section 8.4.

8.2 Control Design

8.2.1 Reference Model

The trajectory is defined through a linear reference model M^* sharing the same dimensions as the low order model M_r; s^* is the reference signal:

$$M^* \ : \begin{cases} x^*_{r_+} = A^*_r x^*_r + B^*_r s^* \\ y^*_r = C^*_r x^*_r \end{cases} \tag{8.7}$$

The trajectory error e_t is defined in the reduced order state space as:

$$e_t = x_r - x^*_r \tag{8.8}$$

The objective is to design a controller such that it minimizes the tracking error e_t. To achieve this objective, an observer-based output feedback controller is designed, that uses both feed-forward and feedback elements.

8.2.2 Observer-based output feedback

The observer design is detailed in section 6.2, the observer equation is given in equation (6.4) and is recalled hereafter:

$$\begin{pmatrix} \hat{x}_{r_+} \\ \hat{D}_{u_+} \end{pmatrix} = \begin{pmatrix} A_r & \tilde{\mathbf{B}}_r \\ 0 & J \end{pmatrix} \begin{pmatrix} \hat{x}_r \\ \hat{D}_u \end{pmatrix} - \mathbf{K_o} C_r (x_r - \hat{x}_r) + \begin{pmatrix} B_r \\ 0 \end{pmatrix} u \tag{8.9}$$

The control law used to perform dynamic trajectory tracking is a PI-like control (L_i, integral part) based on a reference model state x^*_r (L^*) and the estimated states $\hat{x}_r, \hat{d}_u$ (L and L_u).

$$u = -L^* x^*_r - \begin{pmatrix} L & L_u & L_i \end{pmatrix} \begin{pmatrix} \hat{x}_r \\ \hat{d}_u \\ x_i \end{pmatrix} \tag{8.10}$$

The dynamics of the trajectory error e_t writes, from (8.7) and (5.29):

$$\begin{aligned} e_{t_+} =& x_{r_+} - x^*_{r_+} \\ =& A_r x_r + B_r u + B_r d_u + \varepsilon - A^*_r x^*_r - B^*_r s^* \end{aligned} \tag{8.11}$$

The proposed control law leads to the following trajectory error dynamic:

$$\begin{aligned} e_{t_+} =& A_r x_r - B_r (L\hat{x}_r + L^* x^*_r + L_u \hat{d}_u + L_i x_i) + B_r d_u + B^\perp_r \varepsilon \\ & - A^*_r x^*_r - B^*_r s^* \\ e_{t_+} =& A_r (e_t + x^*_r) - B_r (L\hat{x}_r + L^* x^*_r + L_u \hat{d}_u + L_i x_i) \\ & + B_r d_u + B^\perp_r \varepsilon - A^*_r x^*_r - B^*_r s^* \end{aligned} \tag{8.12}$$

As $\hat{x}_r = x_r - e_o = e_t + x_r^* - e_o$, it holds:

$$\begin{aligned} e_{t_+} =& A_r(e_t + x_r^*) - B_r(L(e_t + x_r^* - e_{o_x}) + L^* x_r^* + L_u \hat{d}_u + L_i x_i) \\ &+ B_r d_u + B_r^{\perp}\varepsilon - A_r^* x_r^* - B_r^* s^* \\ e_{t_+} =& (A_r - B_r L)e_t + (A_r - A_r^* - B_r(L + L^*))x_r^* \\ &+ B_r L e_{o_x} + B_r(d_u - L_u \hat{d}_u) - B_r L_i x_i - B_r^* s^* + B_r^{\perp}\varepsilon \end{aligned} \tag{8.13}$$

In (8.13), an adequate choice to recover $d_u - \hat{d}_u$ is to set $L_u = I$ to get:

$$\begin{aligned} e_{t_+} =& (A_r - B_r L)e_t + (A_r - A_r^* - B_r(L + L^*))x_r^* \\ &+ \begin{pmatrix} B_r L & B_r \end{pmatrix} \begin{pmatrix} e_{o_x} \\ d_u - \hat{d}_u \end{pmatrix} - B_r L_i x_i - B_r^* s^* + B_r^{\perp}\varepsilon \\ e_{t_+} =& (A_r - B_r L)e_t + (A_r - A_r^* - B_r(L + L^*))x_r^* \\ &+ \begin{pmatrix} B_r L & \tilde{\mathbf{B}}_r \end{pmatrix} e_o - B_r L_i x_i - B_r^* s^* + B_r^{\perp}\varepsilon \end{aligned} \tag{8.14}$$

The integral term x_i corresponds to:

$$\begin{aligned} x_{i_+} =& x_i + (y_r - y_r^*) \\ x_{i_+} =& x_i + C_r e_t + (C_r - C_r^*)x_r^* \end{aligned} \tag{8.15}$$

Defining $\mathbf{z} = \begin{pmatrix} e_t \\ x_r^* \\ x_i \end{pmatrix}$ and $\mathbf{A_c} = \begin{pmatrix} A_r & A_r - A_r^* & 0 \\ 0 & A_r^* & 0 \\ C_r & C_r - C_r^* & I \end{pmatrix}$, it follows:

$$\begin{aligned} \mathbf{z}_+ =& \left[\mathbf{A_c} - \begin{pmatrix} B_r \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} L & (L + L^*) & L_i \end{pmatrix}\right]\mathbf{z} \\ &+ \begin{pmatrix} B_r L & \tilde{\mathbf{B}}_r \\ 0 & 0 \\ 0 & 0 \end{pmatrix} e_o + \begin{pmatrix} -B_r^* \\ B_r^* \\ 0 \end{pmatrix} s^* + \begin{pmatrix} B_r^{\perp} \\ 0 \\ 0 \end{pmatrix} \varepsilon \end{aligned} \tag{8.16}$$

Now, we can write the closed-loop model including both the observer (see chapter 6, equation (6.4)) and the controller as:

$$\begin{pmatrix} \mathbf{z}_+ \\ e_{o_+} \end{pmatrix} = \mathbf{G} \begin{pmatrix} \mathbf{z} \\ e_o \end{pmatrix} + \tilde{\mathbf{B}}_r^* s^* + \tilde{\mathbf{\Phi}}\varepsilon \tag{8.17}$$

with

$$\mathbf{G} = \begin{pmatrix} \mathbf{A_c} - \begin{pmatrix} B_r \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} L & (L + L^*) & L_i \end{pmatrix} & \begin{pmatrix} B_r L & \tilde{\mathbf{B}}_r \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \\ 0 & \mathbf{A_o} - \mathbf{K_o C_o} \end{pmatrix} \tag{8.18}$$

and

$$\tilde{\mathbf{B}}_{\mathbf{r}}^{*} = \left(\begin{bmatrix} -B_r^* \\ B_r^* \\ 0 \\ 0 \end{bmatrix} \right) \; ; \; \tilde{\mathbf{\Phi}} = \left(\begin{bmatrix} B_r^{\perp} \\ 0 \\ 0 \\ B_r^{\perp} \\ 0 \end{bmatrix} \right) \tag{8.19}$$

The following sections present how to compute the observer and controller gains that stabilize the closed-loop system and minimize the trajectory error. In the general case, finding all-in-one the Lyapunov function, the observer and controller gains result is a non-convex problem. The structure of (8.16) associated with the matrix $\mathbf{G}$ of (8.18) allows to combine a quasi separation principle (solving the observer design apart from the controller design) and ISS properties to solve the problem. Global stability of the closed-loop together with region of convergence are provided. The control algorithm can thus be summarized as:

- step 1 : Computation of the observer gain (see chapter 6).
- step 2 : Computation of the controller gains.
- step 3 : Proof of stability and study of convergence region for the closed-loop algorithm (system (8.17)).

8.3 Computation of Controller Gains

In a similar manner as in 6.2, this section describes how to compute the controller gains based on Input-to-State Stability (ISS). Thereafter, the objective is to compute the controller gains L, L^* and L_i in (8.10) while minimizing the impact

of the modeling and reduction errors on the output tracking error $(y - y^*)$:

$$\begin{aligned}
&\text{Find } L,\ L^*,\ L_i \text{ and } P_t \text{ in } V_t(\mathbf{z}) = \mathbf{z}^T P_t \mathbf{z} \text{ for the model:} \\
&\mathbf{z}_+ = \left[\mathbf{A_c} - \begin{pmatrix} B_r \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} L & (L+L^*) & L_i \end{pmatrix}\right]\mathbf{z} \\
&\qquad + \begin{pmatrix} B_r L & \tilde{\mathbf{B}}_r \\ 0 & 0 \\ 0 & 0 \end{pmatrix} e_o + \begin{pmatrix} -B_r^* \\ B_r^* \\ 0 \end{pmatrix} s^* + \begin{pmatrix} B_r^{\perp} \\ 0 \\ 0 \end{pmatrix} \varepsilon \\
&\text{such that:} \\
&\Delta V_t < -\alpha_t (y - y^*)^T (y - y^*) + \beta_t e_o^T \begin{pmatrix} P_t & 0 \\ 0 & I \end{pmatrix} e_o \\
&\qquad + \gamma_t \varepsilon^T \varepsilon + \omega_t s^{*T} s^* \\
&\text{for given : } (\alpha_t, \beta_t, \gamma_t, \omega_t) > 0 \text{ and } P_t > 0
\end{aligned} \tag{8.20}$$

It holds:

$$\begin{aligned}
(y - y^*) &= \begin{pmatrix} C_r & C_r - C_r^* & 0 \end{pmatrix} \begin{pmatrix} e_t \\ x_r^* \\ x_i \end{pmatrix} = \tilde{\mathbf{C}}_r \mathbf{z} \\
\Rightarrow (y - y^*)^T (y - y^*) &= \mathbf{z}^T \tilde{\mathbf{C}}_r^T \tilde{\mathbf{C}}_r \mathbf{z}
\end{aligned} \tag{8.21}$$

Thus, problem (8.20) is equivalent to:

$$\begin{aligned}
&\left[\ * \ \right] P_t \left[\left(\mathbf{A_c} - \begin{pmatrix} B_r \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} L & (L+L^*) & L_i \end{pmatrix} \right) \mathbf{z} \right. \\
&+ \begin{pmatrix} B_r L & \tilde{\mathbf{B}}_r \\ 0 & 0 \\ 0 & 0 \end{pmatrix} e_o + \begin{pmatrix} -B_r^* \\ B_r^* \\ 0 \end{pmatrix} s^* + \left. \begin{pmatrix} B_r^{\perp} \\ 0 \\ 0 \end{pmatrix} \varepsilon \right] - \mathbf{z}^T P_t \mathbf{z} \\
&< -\alpha_t \mathbf{z}^T \tilde{\mathbf{C}}_r^T \tilde{\mathbf{C}}_r \mathbf{z} + \beta_t e_o^T \begin{pmatrix} P_t & 0 \\ 0 & I \end{pmatrix} e_o + \gamma_t \varepsilon^T \varepsilon + \omega_t s^{*T} s^*
\end{aligned} \tag{8.22}$$

Denoting $X_t = P_t^{-1}$ and using Schur's complement and congruence property with $\mathbf{diag}(X_t, [X_t \quad I], I, I, I)$, condition (8.23) defines a LMI problem corresponding to ISS condition (8.20).

$$\begin{pmatrix} & -X_t & & * & * & * & * & * \\ & 0 & & -\beta_t\begin{pmatrix} X_t & 0 \\ 0 & I \end{pmatrix} & * & * & * & * \\ & 0 & & 0 & -\omega_t I & * & * & * \\ & 0 & & 0 & 0 & -\gamma_t I & * & * \\ \mathbf{A_c} - \begin{pmatrix} B_r \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} L & (L+L^*) & L_i \end{pmatrix} X_t & & \begin{pmatrix} B_r L X_t & \tilde{\mathbf{B}}_r \\ 0 & 0 \\ 0 & 0 \end{pmatrix} & \begin{pmatrix} -B_r^* \\ B_r^* \\ 0 \end{pmatrix} & \begin{pmatrix} B_r^\perp \\ 0 \\ 0 \end{pmatrix} & -X_t & * \\ & \tilde{\mathbf{C}}_r X_t & & 0 & 0 & 0 & 0 & -\alpha_t I \end{pmatrix} < 0 \tag{8.23}$$

8.3.1 Proof of stability for complete low order system

The two previous LMI problems - equation (6.12) from chapter (6) and equation 8.24 - compute the controller and observer gains. This section optimizes the convergence region where the proof of stability is given. Once both the observer and controllers gains are obtained, the matrix $\mathbf{G}$ defined in equation (8.18) is fully defined. The ISS of the complete closed-loop (i.e. controller and observer) is guaranteed if the following problem is solved:

System (8.17) is ISS if it exists a Lyapunov function $V(\mathbf{z}, e_o) = (\mathbf{z}^T \ e_o^T) P \begin{pmatrix} \mathbf{z} \\ e_o \end{pmatrix}$ such that:

$$\Delta V(\mathbf{z}, e_o) < -\alpha(\begin{bmatrix} \|\mathbf{z}\| & \|e_o\| \end{bmatrix}) + \gamma(\|\varepsilon\|) + \omega(\|s^*\|) \tag{8.24}$$

where $\alpha(\|x\|) = \alpha x^T P x$, $\omega(\|s^*\|) = \omega s^{*T} s^*$ and $\gamma(\|\varepsilon\|) = \gamma \varepsilon^T \varepsilon$, with $P > 0$, $\alpha > 0$, $\omega > 0$ and $\gamma > 0$. Then, previous ISS condition is satisfied if:

$$\begin{aligned} (8.24) \Leftrightarrow \begin{bmatrix} * \end{bmatrix} P \begin{bmatrix} \mathbf{G} \begin{pmatrix} \mathbf{z} \\ e_o \end{pmatrix} + \tilde{\mathbf{B}}_r^* s^* + \tilde{\mathbf{\Phi}} \varepsilon \end{bmatrix} - \begin{pmatrix} \mathbf{z} \\ e_o \end{pmatrix}^T P \begin{pmatrix} \mathbf{z} \\ e_o \end{pmatrix} \\ < -\alpha \begin{pmatrix} \mathbf{z} \\ e_o \end{pmatrix}^T P \begin{pmatrix} \mathbf{z} \\ e_o \end{pmatrix} + \gamma \varepsilon^T \varepsilon + \omega s^{*T} s^* \\ \Leftrightarrow \begin{bmatrix} * \end{bmatrix} P \begin{bmatrix} G & \tilde{\mathbf{B}}_r^* & \tilde{\mathbf{\Phi}} \end{bmatrix} - \mathbf{diag}((1-\alpha)P, \omega I, \gamma I) < 0 \end{aligned} \tag{8.25}$$

This defines a generalized eigenvalue problem in $P, \alpha, \gamma, \omega$ and ensures the stability of system (8.24) with the observer gain $\mathbf{K_o}$ and controller gains L, L^* and L_i. Moreover, it ensures that $V(\mathbf{z}, e_o)$ is decreasing for large enough vectors $(\mathbf{z}, e_o)$. The system finally converges to an invariant manifold defined as:

$$\begin{pmatrix} \mathbf{z} \\ e_o \end{pmatrix}^T P \begin{pmatrix} \mathbf{z} \\ e_o \end{pmatrix} < \frac{\gamma \eta^2 + \omega \|s^*\|^2}{\alpha} \tag{8.26}$$

This defines a region of convergence for the vector $(\mathbf{z}, e_o)$. In order to study the convergence of the reduced order state x_r, let us write:

$$\begin{pmatrix} \mathbf{z} \\ e_o \end{pmatrix} = \begin{pmatrix} e_t \\ x_r^* \\ x_i \\ e_o \end{pmatrix} = \begin{pmatrix} I & -I & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{pmatrix} \begin{pmatrix} x_r \\ x_r^* \\ x_i \\ e_o \end{pmatrix} = \mathbf{T} \begin{pmatrix} x_r \\ x_r^* \\ x_i \\ e_o \end{pmatrix} \tag{8.27}$$

Equation (8.26) is also equivalent to:

$$\begin{pmatrix} \mathbf{z} \\ e_o \end{pmatrix}^T P \begin{pmatrix} \mathbf{z} \\ e_o \end{pmatrix} = \begin{pmatrix} x_r \\ x_r^* \\ x_i \\ e_o \end{pmatrix}^T \mathbf{T}^T P \mathbf{T} \begin{pmatrix} x_r \\ x_r^* \\ x_i \\ e_o \end{pmatrix} < \frac{\gamma\eta^2 + \omega\|s^*\|^2}{\alpha} \tag{8.28}$$

In addition, we have:

$$\begin{pmatrix} x_r \\ x_r^* \\ x_i \\ e_o \end{pmatrix}^T \mathbf{T}^T P \mathbf{T} \begin{pmatrix} x_r \\ x_r^* \\ x_i \\ e_o \end{pmatrix} \geq \begin{pmatrix} x_r \\ x_r^* \\ x_i \\ e_o \end{pmatrix}^T \begin{pmatrix} x_r \\ x_r^* \\ x_i \\ e_o \end{pmatrix} \lambda_{min}(\mathbf{T}^T P \mathbf{T}) \tag{8.29}$$

where $\lambda_{min}(A)$ is the smallest eigenvalue of A. It yields:

$$\begin{pmatrix} x_r \\ x_r^* \\ x_i \\ e_o \end{pmatrix}^T \mathbf{T}^T P \mathbf{T} \begin{pmatrix} x_r \\ x_r^* \\ x_i \\ e_o \end{pmatrix} \geq x_r^T x_r \lambda_{min}(\mathbf{T}^T P \mathbf{T}) \tag{8.30}$$

Finally, the invariant manifold with respect to the reduced order state writes:

$$\|x_r\|^2 \leq \frac{\gamma\eta^2 + \omega\|s^*\|^2}{\lambda_{min}(\mathbf{T}^T P \mathbf{T})\alpha} \tag{8.31}$$

A graphical scheme of the convergence region defined by equation (8.31) is shown in figure 8.1. Section 8.4 will give an example of the estimation of this bound as well as the experimental results from the studied robot.

8.3.2 Summary of this closed-loop algorithm

The study of the open-loop system says that for a given workspace in which the norm of the reduced order state x_r is bounded by a given constant c, the modeling error ϵ is bounded by η:

$$\|x_r\|^2 < c \Rightarrow \|\epsilon\| < \eta$$

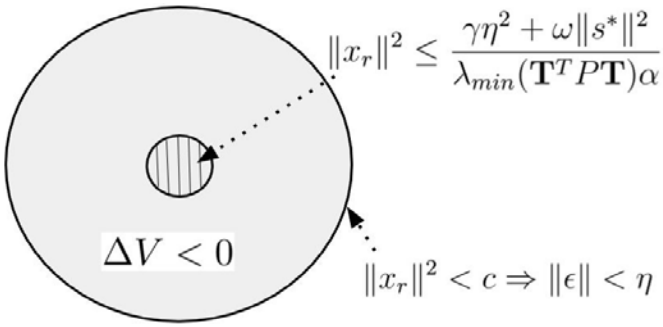


Figure 8.1: *Illustration of convergence region using ISS property. The hatching disk corresponds to the invariant manifold defined in* (8.31)

Then, three LMI constraints problems – equation (6.12) from chapter (6) for the design of the observer, equation 8.24 to design the controller and 8.23 for the stability study – are solved to design a low dimension controller with stability constraint for the full order model. The solutions of these three LMIs guarantee that if the modeling error ϵ is bounded by η, then the reduced order state converges to an invariant manifold defined in (8.26) and (8.31). If this region of convergence is smaller than the workspace in which $\|x_r\|^2 < c$ is satisfied, then the closed-loop system is stable:

Control Design Algorithm:

Study of the open-loop system:

$$\|\epsilon\| < \eta \Rightarrow \|x_r\|^2 \leq \frac{\gamma\eta^2 + \omega\|s^*\|^2}{\lambda_{min}(\mathbf{T}^T P \mathbf{T})\alpha}$$

Solve three LMI constraints problems:

$$\text{If } \frac{\gamma\eta^2 + \omega\|s^*\|^2}{\lambda_{min}(\mathbf{T}^T P \mathbf{T})\alpha} < c$$

Then the closed-loop is stable.

8.4 Experimental Validation

To illustrate the interest and the effectiveness of the methodology proposed, it is tested on soft robotics application. The modeling step is done using the SOFA framework (Coevoet et al. 2017), a unified software to model soft structures thanks to FEM. For this model to be precise, the number of nodes of the mesh has to be significantly high and the model studied is also a large-scale system.

8.4.1 Experimental setup

Experiments are conducted on the trunk-like robot presented in section 3.4.1 whose design is recalled in figure 8.2. It is made entirely of silicone and controlled with 4 cables, actuated by servomotors whose input are the cable lengths. The structure is 18 centimeters long and the thickness at its base and its tip are respectively 2.5 and 1 centimeters. The output of the system is the position of the tip that is measured using a magnetic micro-sensor.

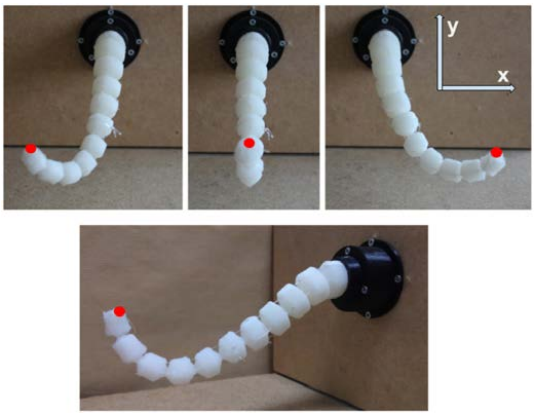


Figure 8.2: *Soft robot used for experimental validation presented in chapter 3.4.1.*

8.4.2 Study of Reduced Order Model

The finite-element mesh of this robot is made of 1557 nodes. The dimensions of the state vector is also $1557 \times 3 \times 2 = 9342$ state variables (3 directions of space for displacement and velocity). Then, model order reduction provides us with a low-order system of dimension 6. As the computational cost of the $\mathcal{H}_\infty$ norm is high, a common way to measure the accuracy of the reduced order system is to measure the $\mathcal{H}_2$ norm error between the full and reduced order models:

$$\|M - M_r\|_{\mathcal{H}_2} \tag{8.32}$$

For this work we use the moment matching reduction method, implemented within the MORE toolbox (Poussot-Vassal & Vuillemin 2012). Table 8.1 gathers the comparison of the $\mathcal{H}_2$ errors for different size of reduced systems, provided that the norm of the full order model $\|M\|_{\mathcal{H}_2} = 0.42$. The best result regarding the $\mathcal{H}_2$ error is achieved for reduced order systems of dimension 26 and higher, they present an error of 24.67% compared to the full order model. The $\mathcal{H}_2$ error between the full order system and the reduced order system of dimension 2 is 25.4% wich is close to the optimal solution. Indeed, the $\mathcal{H}_2$ error decreases from

Dimensions of M_r	2	4	6	10	26	30	100
$\mathcal{H}_2$ norm error $\|M - M_r\|_{\mathcal{H}_2}$ in %	25.40	24.70	24.69	24.69	24.67	24.67	24.67

Table 8.1: *Comparison of the $\mathcal{H}_2$ norm error between the full and reduced order systems for different dimensions of reduced systems.*

a system with dimension 2 to dimension 6 and then converges slowly to the optimal solution. A good compromise between the accuracy and the complexity of the low order model is to choose a reduced order system of dimension 6.

Bound η of the modeling error

By running multiple simulations, we cover a workspace of the robot that is considered as exhaustive (we cover the whole range of possible actuation) and we get the maximal values of the norms of the full and reduced order states:

$$\|x\|^2 = 2.39\ 10^8 \text{ and } \|x_r\|^2 = 2.37\ 10^8 \tag{8.33}$$

In this workspace, and from (8.6), the norm bound of the modeling error is:

$$\eta = 4.19\ 10^3 \tag{8.34}$$

Invariant manifold for the system studied

The invariant manifold defined in equation (8.31) and depicted in figure 8.1 is defined with the following parameters:

$$\begin{aligned} \alpha &= 0.05 \\ \gamma &= 10^{-2} \\ \omega &= 2.10^{-3} \end{aligned} \tag{8.35}$$

For the case where $s^* = 0$, i.e. the robot comes back to its rest shape, the invariant manifold writes:

$$\|x_r\|^2 \leq \frac{\gamma\eta^2}{\lambda_{min}(\mathbf{T}^T P \mathbf{T})\alpha} = 9.14\ 10^6 \tag{8.36}$$

In this case, the invariant manifold corresponds to 3.87% of the robot workspace. Therefore, for an initial condition that satisfies (8.33), the system is stable and converges to the invariant manifold.

8.4.3 Validation of control design

Experiments are done on the real-world setup of the cable-driven presented in section 3.4.1, they are conducted with the same sampling time as for the modeling process; point-to-point control and trajectory tracking experiments are conducted.

Point-to-Point control

The reference is set such that the robot starts from its rest position, converges to a first deformed position and then to a second deformed position. Results are presented in figures 8.3 and 8.4. It shows a diminution of oscillations, a faster time response and a cancellation of the static error compared to the open-loop.

Figure 8.3: *Point-to-point closed-loop control. Top: output along x-axis, Bottom: output along y-axis. Black: Reference signal. Red: Output of reference model and Blue: Robot's end-effector's displacement.*

Sinusoidal trajectory

The second set of experiments consists of tracking a sinusoidal signal along the x-axis of the robot; the reference for axis y is zero. Results are presented in figure 8.5. The behavior along the first axis follows accurately the trajectory while the second outputs oscillates around the reference signal. Of course, this residual oscillation is due to the physical coupling between the actuators.

The mechanical coupling between the actuators limits the performances of the control algorithm as shown in 8.5. Moreover, the friction between the cables and the structure also limits the performances of the closed-loop algorithm, this requires deepening the actuation system to remove this constraint. In addition, the workspace of the robot is limited approximately to a circle of

Figure 8.4: *Point-to-point closed-loop control. Top: output along x-axis, Bottom: output along y-axis. Black: Reference signal. Red: Output of reference model and Blue: Robot's end-effector's displacement.*

Figure 8.5: *Closed-loop trajectory tracking. Top: output along x-axis, Bottom: output along y-axis. Black: Reference signal. Red: Output of reference model and Blue: Robot's end-effector's displacement.*

radius 6 centimeters around the rest position of the robot. If the reference trajectory is set outside of this workspace, the performances start to decrease. This is not a strong limitation as 6 centimeters represent 33.3% of the length of the structure. This workspace limitation is explained by the linearization assumption. For larger deformation, the use of a nonlinear model may be necessary; this is an ongoing research topic.

Part V

Perspectives and Conclusions

Introduction

For now, the validated control laws are all based on linear models. This is a known limitation of this work as it constrains the guaranteed domain of the controller. Tracks to tackle this issue are highlighted in chapter 9 to extend the results to nonlinear models. The idea is to linearize the large-scale nonlinear model around several operating points and interpolate between these points to cover a wider workspace.

The definition of good performances for soft robots is subject to contrasts. On the one hand, the target is to enable safe interactions with humans or the environment in general. Thus, it is crucial that soft robots exhibit high compliance behavior. On the other hand, the objective is to guarantee increased effectiveness in the robots motions. This second target is solved designing a dedicated dynamic controller for the robot. However many controllers impose a reduction of the robots compliance to achieve specified performances, and thus lose the interet of soft robots; this dilemma is discussed in details in (Della Santina et al. 2017). It is also crucial to take into account these potential contacts into the design of the control law; this future work is discussed in chapter 10.

Finally, chapter 11 concludes this manuscript by giving a summary of the contributions of this work and highlighting remaining questions to be solved related to dynamic control of soft robots.

Chapter 9

Increased Guaranteed Domain Of Dynamic Controller: Towards Nonlinear Models

In this manuscript, all the results applied to real robotics platform are based on linear models. In other words, the stability and performances are guaranteed in a given neighborhood around an equilibrium point. This is a known limitation of this work and this chapter presents tracks to overcome this bottleneck.

To extend the guaranteed domain of the control algorithms, a solution is to design a controller valid for the nonlinear system, i.e. without assumption about area of validity. Let us recall that the dynamic behavior of soft robots is given by the second law of Newton:

$$\mathbf{M}(q)\ddot{q} = \mathbf{P}(q) - \mathbf{F}(q,\dot{q}) + \mathbf{H}(q)^T u \tag{9.1}$$

Defining the state vector $x = (\dot{q}, q)^T$, the previous dynamic equation can be written via a control-affine non-linear state-space equation:

$$(NLS) \ : \ \dot{x} = A(x)x + B(x)u \tag{9.2}$$

This defines a non-linear large-scale state-space system. While it is already challenging to design a controller for nonlinear systems, the complexity of the problem increases with the dimensions of the system.

This chapter presents tracks to extend the workspace of the robot while keeping the complexity of the robot small enough to make the design of a controller tractable.

Contents

9.1 Introduction: basic concepts of LPV systems 104
9.2 Collection of Linear Systems 105
9.3 Linear Parameter Varying (LPV) Models 106
9.3.1 Radial Basis Functions 106
9.3.2 Controller Design based on LPV Models 108
9.4 Discussion and Future Work 109

9.1 Introduction: basic concepts of LPV systems

Linear Parameter varying (LPV) systems have gained popularity during the 1990s, benefiting from the extension of $\mathcal{H}_\infty$ optimal control. Many nonlinear systems can be written as quasi-LPV systems, which has two advantages: the first one is to avoid writing a nonlinear model that requires precise knowledge of the process studied. The second one is to take advantage of all the techniques developed for LTI systems.

Quasi-LPV systems are of the form:

$$\begin{cases} \dot{x} = A(\rho(t))x(t) + B(\rho(t))u(t) \\ y = C(\rho(t))x(t) + D(\rho(t))u(t) \end{cases} \tag{9.3}$$

Polytopic systems are a common way of modeling LPV systems (Geromel & Colaneri 2006*a*, Apkarian & Tuan 2000). This kind of models writes:

$$\begin{bmatrix} A(\rho(t)) & B(\rho(t)) \\ C(\rho(t)) & D(\rho(t)) \end{bmatrix} = \sum_{i=1}^{N} \rho_i(t) \begin{bmatrix} A_i & B_i \\ C_i & D_i \end{bmatrix} \tag{9.4}$$

The birth of LPV systems comes from gain scheduling techniques (Tóth 2010), where the idea is to linearize nonlinear systems around different operating points yielding to a collection of local LTI models. Then, interpolation functions link each local subsystems. These interpolation functions are called scheduling function and to describe the change of operating point, the scheduling signal is used and referred to as ρ. Therefore, the resulting controller are dependent on the varying signal ρ, thus resulting in *parameter varying* systems.

9.2 Collection of Linear Systems

A solution of equation (9.2) going through x_0 at t_0 is denoted by $\phi(t, t_0, x_0)$. Then, $x_e \in \mathbb{R}^n$ is an equilibrium point for equation (9.2) (NLS) if the solution ϕ is defined and verifies $\phi(t, 0, x_e) = x_e, \ \forall t$.

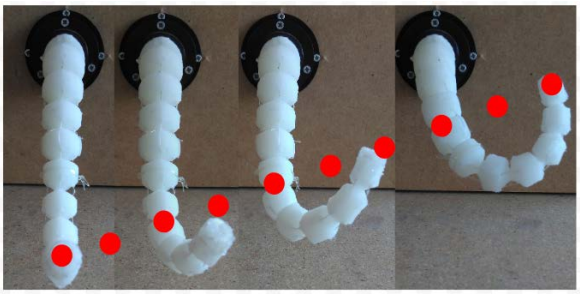


Figure 9.1: *Trajectory along which the nonlinear model is linearized.*

Let us consider a fixed number $\mathcal{N}$ of equilibrium points $x_{e_i}, \ i \in \{1, \ldots, \mathcal{N}\}$ along a trajectory shown in figure 9.1. These equilibrium points are induced by the gravity field and a collection of inputs u_i, such that:

$$0 = A(x_{e_i}) + B(x_{e_i})u_i \tag{9.5}$$

Around each of the equilibrium point x_{e_i}, one can linearize the non-linear equation (NL), yielding to a collection of linear systems:

$$\dot{x} = A_i x + B_i u, \ i \in \{1, \ldots, \mathcal{N}\} \tag{9.6}$$

The POD reduction algorithm is well suited for nonlinear systems. The snapshots are captured so that the entire workspace of the robot is modeled, in other words all the subsystems (A_i, B_i) are included in the snapshot space. The reduction provides the projection matrices V_r and W_r that are valid for all of the subsystems. For the entire workspace of the robot, the projection writes:

$$x_r = W_r x \ ; \ x \approx V_r x_r \tag{9.7}$$

The keypoint of using the POD algorithm, is that all subsystems (A_i, B_i) share the same projectors, yielding to a collection of low order linear systems:

$$\dot{x}_r = A_{ri} x_r + B_{ri} u, \ i \in \{1, \ldots, \mathcal{N}\} \tag{9.8}$$

Once these low order systems computed, different strategies to design a controller exist. An easy way to design a controller is to design it directly for all

the subsystems. If all the subsystems (A_{ri}, B_{ri}) share a common decreasing Lyapunov function, then system (9.8) is stable, i.e. if there exist a matrix $P_r \in \mathbb{R}^{r\times r}$ such that $P_r > 0$ and

$$(*) + P_r A_{ri} - P_r B_{ri} F < 0, \forall i \in \mathcal{I} \tag{9.9}$$

then the control law $u = -Fx_r$ makes system (9.3) stable.

The solution using this representation provides an efficient and easy-to-implement controller but does not study the system behavior between two equilibrium points; this is achieved by studying LPV systems in next section.

9.3 Linear Parameter Varying (LPV) Models

A second way to design controllers is to use Linear Parameter Varying (LPV) models. The method consists of interpolating between all the linear models to find a LPV system using any interpolating basis of functions with adequate properties. In our case we use radial basis functions.

9.3.1 Radial Basis Functions

Let us study the evolution of the coefficient of the reduced order system matrix A_r of the cable-driven soft robot (see figure 3.1). The reduction is done using POD algorithm and the reduced order system is of dimension 6, the matrix A_r writes:

$$A_{r_i} = \begin{pmatrix} a_{11_i} & a_{12_i} & \dots & a_{16_i} \\ a_{21_i} & \ddots & \ddots & a_{26_i} \\ a_{61_i} & \dots & a_{65_i} & a_{66_i} \end{pmatrix} \tag{9.10}$$

The evolution of these three coefficients along the trajectory defined in figure 9.1 is shown in figure 9.2. The evolution of the coefficients through time suggests that it is possible to use interpolating techniques to reconstruct / approximate them. According to the nice properties of radial basis functions, such as sparse universal approximation, these first trials are focused on these functions.

A radial basis function is a real-valued function ϕ whose value depends only on the distance from the origin, so that $\phi(x) = \phi(\|x\|)$; or alternatively on the distance from a given point c, so that $\phi(x, c) = \phi(\|x - c\|)$. This point c is then called a center. Gaussian function of the form $\phi(x) = e^{-x^2}$ are commonly used.

A radial basis network is created to approximate a function defined by a set of data points, in our case the linear subsystems. A radial basis network is a network with two layers, a hidden layer of radial basis neurons and an output layer of linear neurons.

Figure 9.2: *From left to right : evolution of coefficient* a_{11}, a_{21} *and* a_{33} *of matrix* A_r *from equation* (9.10).

The inputs of the network are the coefficient of the systems matrices (A_{ri}, B_{ri}) for each subsystems, the networks output is the function that approximates those coefficient that writes:

$$\lambda_i(w) = e^{-(b_i\|w-c_i\|)^2} \tag{9.11}$$

where b_i are the bias values, w are the input weight values and c are the centers values.

From this approximated function, we get the following system:

$$\dot{x}_r = \sum_{i=0}^{N} \lambda_i A_{r_i} x_r \tag{9.12}$$

However, the convex sum does not hold, i.e. $\sum_{i=0}^{N} \lambda_i \neq 1$; yet it is required to design a controller for the LPV systems.

The sector nonlinearity approach is a systematic way to derive a polytopic model. Values of λ_i are bounded and it holds $\lambda_i \in [0,1] = [\underline{\mathrm{m}}, \bar{m}]$, one can also write $\lambda_i = \omega_1^i \underline{\mathrm{m}} + \omega_2^i \bar{m}$, where:

$$\omega_1^i = 1 - \lambda_i \quad ; \quad \omega_2^i = \lambda_i \tag{9.13}$$

and the function ω_i^i satisfy the convex sum property: $\omega_1^i + \omega_2^i = 1$ and $\omega_i^i \geq 0$. Let us define functions h_i:

$$h_{1+i_0+i_1\times 2+\ldots+i_{k-1}\times 2^k} = \prod_{j=1}^{N} \omega_{i_j}^j \tag{9.14}$$

We finally end with a polytopic model (9.15), based on which the design of a controller is detailed in next section.

$$\dot{x}_r = \sum_{i=0}^{N} h_i A_i x_r \tag{9.15}$$

Remark: *The number of linear models grows exponentially with the number of nonlinear functions.*

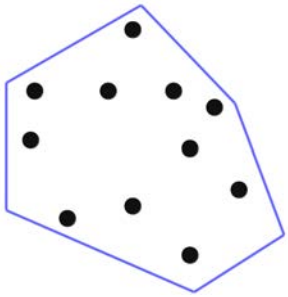


Figure 9.3: *Convex hull of a set of points in the plane.*

9.3.2 Controller Design based on LPV Models

From previous section, radial basis function together with the sector nonlinearity approach provides us with a polytopic model (9.15). The term polytopic comes from the fact that h evolves over the unit simplex defined by:

$$\Gamma : \left\{ \sum_{i=0}^{N} h_i = 1 \ ; \ h_i \geq 0 \right\} \tag{9.16}$$

The polytope Γ can be defined from the set of its vertices, the notion of convex hull is illustrated in figure 9.3.

Proposition of (Boyd et al. 1994):

The LPV polytopic system (9.15) *is quadratically stable if and only if there exists a matrix* $P = P^T > 0$ *such that:*

$$A_{r_i}^T P + P A_{r_i} < 0 \tag{9.17}$$

hold for all $i = 0, \ldots, N$.

Condition (9.17) corresponds to a set of N LMI constraints that, depending on the number of vertices of the polytope, can sometimes lead to a huge number of constraints to solve. The same problem as for switched systems (described in chapter 10) appears as a common matrix P satisfying the LMIS (9.17) may not exist even if all matrices A_i are stable and the polytope is stable.

As the number of parameters N growths, so does the complexity of the condition to satisfy. It can also be time and memory consuming as the number of LMIs to be solved is 2^N. In the discrete-time case, nonquadratic Lyapunov functions may be used to derive stabilization conditions for nonlinear systems using LPV representation (Kruszewski et al. 2008).

The simplest control design would be a linear state feedback controller $u = -Lx_r$ corresponding to the closed-loop system:

$$x_{r_+} = \sum_{i=0}^{N} h_i (A_{r_i} - B_{r_i} L) x_r \tag{9.18}$$

In the literature, a commonly used control law is the parallel distributed compensation (PDC), it is composed of linear feedbacks assembled together with the same nonlinear function h_i as in the model:

$$u = -\sum_{i=1}^{N} h_i L_i x \tag{9.19}$$

9.4 Discussion and Future Work

The previous section has presented a low order LPV polytopic system. As for the previous chapter, the objective is to design a control law based on this reduced dimensions approximated system and apply it to the large-scale model.

To study the robustness of the control law, one should study the modeling errors coming from the reduction algorithm. This is made possible for linear system using POD algorithm. Its extension to LPV models has to consider a global error coming from the convex aggregation of the polytopic vertices. This extension is left for future research.

Chapter 10

Dynamic Control with Contacts with the Environment

Wherever there is need for interactions between robots and men, integrating soft materials into the robot can only make this interaction safer. There is also a need to be able to control these soft robots while dealing with potential contacts with their environment.

An optimization-based open-loop strategy based on real-time inverse problem has been presented in (Coevoet, Escande & Duriez 2017). In addition, authors of (Della Santina et al. 2018) have presented a dynamic control strategy dealing with contacts, based on piece-wise constant curvature model.

In addition to the dimensionality issues mentioned in this manuscript, modeling contacts implies solving non-smooth dynamical equations. Modeling and control non-smooth mechanical systems is still an open problem, even if some solutions exist (Brogliato 2016). This chapter presents tracks to design a dynamic control method dealing with contact based on a finite element model.

Contents

10.1 Dynamic Model with contacts 112
10.2 Controller Design using Switched Systems Theory 113
10.3 Inverse Simulation-based Controller 114
10.4 Summary and Future Work 115



Figure 10.1: *Illustration of Signorini's law for contacts given in equation* (10.3).

10.1 Dynamic Model with contacts

One of the key-point of studying soft robots is to consider contacts with the environment. This would for instance enable safe human/robot interactions. Contacts between robots and the environment may be seen as a disturbance applied to the robot. In this case, part of the methods presented in this manuscript applies. But what happens if the contact changes the direction of actuation? In this case, the model would change over time.

From chapter 3, the dynamics of a soft robots without studying contacts writes:

$$\mathbf{M}(q)\ddot{q} = \mathbf{P}(q) - \mathbf{F}(q,\dot{q}) + \mathbf{H}^T(q)u \tag{10.1}$$

Considering contacts, the non-smooth dynamic model writes:

$$\mathbf{M}(q)\ddot{q} = \mathbf{P}(q) - \mathbf{F}(q,\dot{q}) + \mathbf{H}_a^T(q)u - \mathbf{H}_c^T\lambda_c \tag{10.2}$$

where $\mathbf{H}_c(q)$ is a matrix containing the directions of contacts and λ_c their intensities.

Contacts are usually formulated as a complementarity condition:

$$0 \leq \lambda_c \perp \delta_c \geq 0 \tag{10.3}$$

where δ_c is the gap between two colliding points, see figure 10.1.

Considering the output as the distance δ_c, this expression can be put in a standard state-space representation:

$$\begin{aligned} \dot{x} &= Ax + Bu + F\lambda_c \\ y &= Cx \\ 0 &\leq \lambda_c \perp y \geq 0 \end{aligned} \tag{10.4}$$

Author of (Vieira 2018) shows how to write this Linear Complementary System (LCS) as a hybrid dynamical system. He points out that it is not the most

efficient way to study this kind of systems but hybrid systems are widely used in control theory and this framework offers important tools for analysis.

Section 10.2 presents a method to study the stability and the design of a closed-loop controller in presence of contacts using the theoretical framework of switched systems. Then, section 10.3 studies a way to extend previous results about quasi-static controller to the dynamic case with contacts handling.

10.2 Controller Design using Switched Systems Theory

Hybrid systems are the association of a finite set of dynamical systems and a switching law that indicates at each time which mode is active (Liberzon 2003). It is of the form:

$$\dot{x} = f_\sigma(x, u, t) \tag{10.5}$$

where $x \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}^m$ is the input and $\sigma \in \mathbb{R} \to \mathcal{I}_n$ is the switching law and $\mathcal{I}_N = \{1, ..., N\}$. In the linear case, system (10.5) writes:

$$(HS) : \dot{x} = A_\sigma x + B_\sigma u \tag{10.6}$$

When dealing with hybrid systems, considering discrete-time switched systems has many advantages: avoid well-posedness of solutions (Filippov solutions etc.), avoid Zeno phenomenon etc. We also consider open-loop discrete-time hybrid systems in the form:

$$x_+ = A_\sigma x \tag{10.7}$$

If all the modes share a common Lyapunov function, then the switched system is stable (Geromel & Colaneri 2006*b*), i.e. for a quadratic Lyapunov function $V(x) = x^T P x$, if there exists a matrix $P \in \mathbb{R}^{\kappa \times \kappa}$ such that $P > 0$ and

$$A_i^T P A_i - P < 0, \forall i \in \mathcal{I} \tag{10.8}$$

then the switched system (10.6) is globally uniformly stable.

However, system (10.7) may be stable without finding feasible solution for the previous LMI problem. Considering multiple Lyapunov function reduce the conservatism of the solution. Let us consider Lyapunov function defined as $V(\sigma, x) = x^T P_\sigma x$.

If there exist P_i, $i \in \mathcal{I}$, such that $P_i > 0$ and

$$A_i^T P_j A_i - P_i < 0, \forall (i, j) \in \mathcal{I}_N^2 \tag{10.9}$$

then the discrete-time switched system (10.6) is globaly uniformaly stable (Daafouz et al. 2002).

In the case where some of the modes A_i are unstable, the problem is to design a controller that stabilizes system (10.6).

Let us recall the class of Metzler matrices denoted by $\mathcal{M}$ and constituted by all matrices $\Pi \in \mathbb{R}^{n\times n}$ with elements π_{ij} such that:

$$\pi_{ij} \geq 0, \ \sum_{i=1}^{N} \pi_{ij} = 1, \ \forall i, j \tag{10.10}$$

If the Lyapunov-Metzler inequalities

$$\sum_{j\in\mathcal{I}_N} \pi_{ij} A_i^T P_j A_i - P_i < 0, \forall i \in \mathcal{I}, P_i > 0 \tag{10.11}$$

holds then the following switching law

$$\sigma(x) = arg \ \min_{i\in\mathcal{I}_N} x^T P_i x \tag{10.12}$$

makes the origin of the switched system (10.6) asymptotically stable (Geromel & Colaneri 2006*b*).

10.3 Inverse Simulation-based Controller

The simulation framework SOFA - used previously to build the dynamic model - could also be used to control the robot while handling its interactions with the environment, provided that it is modeled too.

Author of (Morales Bieze 2017) presents a closed-loop controller for soft robots under quasi-static assumptions. It uses the SOFA framework to control the robot via an inverse kinematics simulation and a forward kinematics simulation used as an observer as shown in figure 10.2. The controller is tuned based on a quasi-static model of the robot.

In addition, also based on these simulation tools, authors of (Coevoet, Escande & Duriez 2017) present an open-loop algorithm to drive the robot in presence of contacts. The algorithm consits of solving an optimization problem formulated as a quadratic problem (QP) at each time step.

The remaining challenge is to assemble the parts together to design a closed-loop dynamic controller while handling contacts. In the dynamic case, the inverse model would permit to compute the control input to apply to reach a desired $\ddot{q}$. For the quasi-case, a study of the controller robustness has been conducted in (Morales Bieze et al. 2018). It shows that the designed controller is robust with respect to large modeling errors, as long as the sign of the estimated Jacobian matrix is correct.

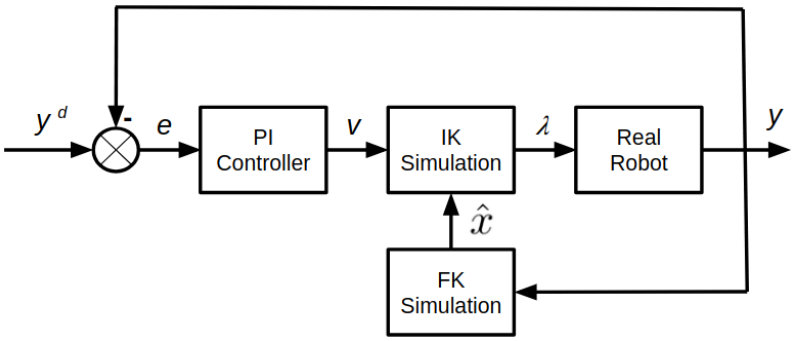


Figure 10.2: *Closed-loop control design based on kinematics simulations of the robot, from (Morales Bieze 2017).*

A similar study should be done for the dynamic case to study the robustness of the dynamic controller. In addition, the presence of contacts slows down the simulation and thus decreases the performances. Including a model order reduction algorithms into the design of the inverse problem would help to reach real-time performances.

10.4 Summary and Future Work

The ability to control the robot while handling contacts with its environment is mandatory to take full benefit of soft robots compliance. This challenge raises theoretical questions about control of non-smooth dynamical systems (Brogliato 2016). As the contacts may lead to high deformations of the robot structure, a nonlinear model of the robot is necessary to achieve this task.

In addition to the theoretical bottleneck, the hardware implementation is also challenging. Either to avoid the contacts or to use it to perform the task, it requires to detect the contact. This requires either an external sensing hardware or the ability to integrate it into the robots structure to be completely independent.

Chapter 11

General Conclusion

This thesis manuscript provides new methods to model and control soft robots, without any particular assumptions on the robots geometry. It covers the entire development of the controller, from the modeling step to the practical experimental validation.

From a theoretical point a view, large-scale dynamical systems along with model reduction algorithms are studied. Indeed, modeling soft structures implies solving equations coming from continuum mechanics. This can be done using different algorithms and in this work we opt for the finite element method. This method requires a spatial mesh of the robot that needs to be precise to obtain a precise model. This leads to work with models of large dimensions, that are not suitable to design control algorithms.

First, this work presents a controller designed for the large-scale model using the knowledge of an open-loop Lyapunov function. Starting from an extension of the energy formulation of the structure, a large-dimension state-feedback control law is designed to tune the performances of the closed-loop.

Then, this work investigates model reduction algorithms to design low dimension controllers and observers to drive soft robots. In both case, reduction errors are taken into account to cope with modeling uncertainties and provide maximum robustness. Two methods are proposed, the first one based on the existence of a large-scale open-loop Lyapunov function, the second method is based on Input-To-State Stability analysis. In all cases, the objective is to design

low-dimensions controllers with stability and performances guaranteed for the large-scale systems.

In addition to these theoretical studies, different experimental setups are used to illustrate the results. A cable-driven soft robot and a pressurized soft arm are used to test the control algorithms. Through these different setups, we show that the method can handle different types of actuation (cables or pressure), different geometries and mechanical properties. This emphasizes one of the interests of the method, its genericity.

Several work perspectives have been identified and detailed in chapters 9 and 10 but many issues still need to be resolved. For theoretical studies, the extension of this work to LPV models raises the question of the complexity regarding the model reduction algorithms. Indeed, model reduction applied to nonlinear systems is an active and promising field of research where many questions need to be answered.

Remember that the problems have been written as LMI constraints problems using several pessimistic options (energy-based Lyapunov function, linear control and observer, choices for bounds...) rendering sufficient conditions only. Of course, if the LMI problems written along this work do not give solution, fallback solutions have to be thought, either relaxing the pessimistic choices made or changing the design of the reduced model or even modifying the design of the robot. In addition to the design of control algorithms, the FEM models presented in this work has many applications in this regard. Authors of (Morzadec et al. 2019) present an optimization strategy to design the shape of soft structures based on a FEM model. This method with control objectives would be similar to morphological computation techniques and could help to embed the control algorithm into the soft structure. Still based on a FEM model, (Zheng et al. 2019) presents an a priori controllability verification method.

Finally, some questions related to dynamic control of soft robots are still opened. First, the definition of performances for soft robots should be discussed, as in (Della Santina et al. 2017). There is indeed a potential conflict between the highly compliant behavior of the structure and the specifications imposed on the systems. In addition, a comparison between existing methods needs to be done: model-free control, model based on PCC assumptions and FEM model-based controller. A unified framework for these approaches could also be investigated. In addition, only *manipulator-like* robots have been considered in this work but extending this study to mobile robotics platforms to perform dynamic locomotion task using soft materials robots is also an interesting challenge.

List of Publications

Journal Articles

Trajectory Tracking Control Design for Large Scale Linear Dynamical Systems with applications to Soft Robotics
M. Thieffry, A. Kruszewski, T.M. Guerra, C. Duriez
accepted for publications in Transactions on Control Systems Technology

(Thieffry et al. 2019) *Control Design for Soft Robots based on Reduced Order Model*
M. Thieffry, A. Kruszewski, C. Duriez, T.M. Guerra
Robotics and Automation Letters, 2019

(Coevoet et al. 2017) *Software Toolkit for Modeling, Simulation, and Control of Soft Robots*
E. Coevoet, T. Morales, F. Largilliere, Z. Zhang, M. Thieffry, M. Sanz, B. Carrez, D. Marchal, O. Goury, J. Dequidt, C. Duriez
Advanced Robotics, journal of the Robotics Society of Japan, 2017

Conference Papers

(Katzschmann, Thieffry et al. 2019) *Dynamically Closed Loop Controlled Soft Robotic Arm using a Reduced Order Finite Element Model with State Observer*
R. Katzschmann, M. Thieffry, A. Kruszewski, O. Goury, T.M. Guerra, C. Duriez, D. Rus
International Conference on Soft Robotics - RoboSoft19

(Thieffry et al. 2018) *Reduced Order Control of Soft Robots with Guaranteed Stability*
M. Thieffry, A. Kruszewski, T.M. Guerra, C. Duriez
European Control Conference - ECC18

(Thieffry et al. 2017) *Dynamic Control of Soft Robots*
M. Thieffry, A. Kruszewski, O. Goury, T.M. Guerra, C. Duriez
IFAC World Congress, 2017

APPENDIX

A

Preliminary in Systems Theory and Control

A.1 Introduction

The larger class of finite dimensional systems is the class of nonlinear systems. These systems are defined as:

$$\dot{x}(t) = f(x(t), u(t)) \tag{A.1}$$

On the other hand, the most easy way to model dynamical system is through Linear Time Invariant (LTI) systems. In this case, the dynamics of system (A.1) writes:

$$\dot{x}(t) = Ax + Bu(t) \tag{A.2}$$

where A and B are constant matrices.

Going from nonlinear to linear models, different levels of complexity exists to model dynamicl model. The first extension of LTI systems is the use of Linear Time Varying systems (LTV), and its direct generalization, the Linear Parameter Varying (LPV) systems. These systems are written as:

$$\dot{x}(t) = A(\rho(t))x + B(\rho(t))u(t) \tag{A.3}$$

In addition, it is possible to approximate nonlinear systems by another class of systems called quasi-LPV systems. This classification is recalled in figure A.1.

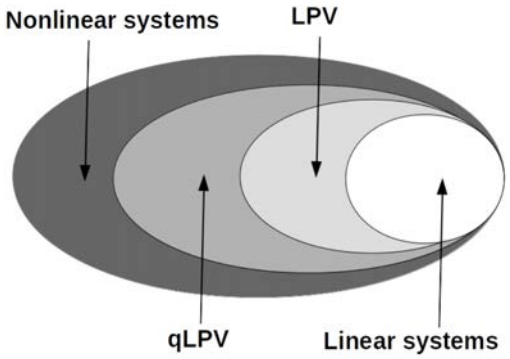


Figure A.1: *Classification of finite dimensional dynamical systems*

A.2 Properties of LTI Systems

A.2.1 Gramians

Considering a LTI system, let us define its controllability $\mathcal{P}$ gramian.

$$\mathcal{P} = \int_0^\infty e^{A\tau} B B^T e^{A^T \tau} d\tau \tag{A.4}$$

It can be found as the solution of the following Lyapunov equation:

$$A\mathcal{P} + \mathcal{P}A^T = -B^T B \tag{A.5}$$

In the same manner, the observability gramian $\mathcal{O}$ is defined as:

$$\mathcal{O} = \int_0^\infty e^{A\tau} C C^T e^{A^T \tau} d\tau \tag{A.6}$$

with the corresponding Lyapunov equation:

$$A\mathcal{O} + \mathcal{O}A^T = -C^T C \tag{A.7}$$

The Hankel singular values of a LTI system are defined as the square roots of the eigenvalues of the product of the controllability and observability gramians.

$$\sigma_i = \sqrt{\lambda_i(\mathcal{P}\mathcal{O})} \tag{A.8}$$

These Hankel singular values are the basis of the balanced truncation, which is described in section hereafter.

A.2.2 Norms of systems

A.2.2.1.0 $\mathcal{H}_2$ norm

The $\mathcal{H}_2$ norm of a dynamical linear system M, whose transfer function is $M(j\omega)$, is defined as:

$$||M||^2_{\mathcal{H}_2} = \frac{1}{2\pi}\int_{-\infty}^{\infty} trace(M(j\omega)M(-j\omega)^T)d\omega \tag{A.9}$$

Using the definition of gramians presented above, it holds:

$$||M||^2_{\mathcal{H}_2} = trace(C\mathcal{P}C^T) = trace(B^T\mathcal{O}B) \tag{A.10}$$

The $\mathcal{H}_2$ norm is widely used in model reduction algorithms to study the accuracy of the reduced model.

A.2.2.2.0 $\mathcal{H}_\infty$ norm

The $\mathcal{H}_\infty$ of a MIMO LTI system is the maximum singular values of the transfer function across all frequencies. It is defined as:

$$||M||_{\mathcal{H}_\infty} = \max_{\omega\in\mathbb{R}} \sigma_{max}(M(j\omega)) \tag{A.11}$$

The $\mathcal{H}_\infty$ norm is commonly used in control theory to study the robustness of the controller. However, computing the $\mathcal{H}_\infty$ norm of large-scale systems is a complex task and designing a reduction algorihm that minimize the $\mathcal{H}_\infty$ norm between the full and low order systems is still an open problem.

Few solutions exists to handle this problem (Megretski 2006, Vuillemin et al. 2014, Benner & Mitchell 2018, Benner et al. 2018).

A.2.3 Moments

Consider a system with m inputs and p outputs with transfer function $H(s)$. It can be decomposed through a Laurent series expansion around a given shift point $\sigma \in \mathbb{C}$ as:

$$H(s) = \sum_{i=0}^{\infty} \eta_i(\sigma)\frac{(s-\sigma)^i}{i!} \tag{A.12}$$

$\eta_i(\sigma) \in \mathbb{C}^{p\times m}$, is the i-th moment of $H(s)$ at σ associated to the model and is defined as

$$\eta_i(\sigma) = (-1)^i \frac{d^i H(s)}{ds^i}\bigg|_{s=\sigma} \tag{A.13}$$

A.2.4 Lyapunov Stability

A solution of equation $\dot{x} = f(x)$ going through x_0 at t_0 is denoted by $\phi(t, t_0, x_0)$. Then, $x_e \in \mathbb{R}^n$ is an equilibrium point for this equation if the solution ϕ is defined and verify $\phi(t, 0, x_e) = x_e, \ \forall t$.

Let us consider $x_e = 0$, a Lyapunov candidate function V that verifies $V(0) = 0$ is a definite positive function satisfying

$$\begin{aligned} &\|x\| \to \infty \Rightarrow V(x) \to \infty \\ &\alpha(\|x\|) \leq V(x) \leq \beta(\|x\|) \end{aligned} \tag{A.14}$$

where α and β are positive definite functions.

A system is said to be globally asymptotically stable according to the initial conditions if it exists a Lyapunov candidate function such that:

$$\lim_{t \to \infty} V(x(t)) = 0 \ , \quad \forall x(0) \in \mathbb{R}^n \tag{A.15}$$

A.2.5 Linear Matrix Inequality (LMI)

A linear matrix inequality (LMI) has the form

$$F(x) = F_0 + \sum_{i=1}^{m} x_i F_i > 0 \tag{A.16}$$

where $x \in \mathbb{R}^m$ is the decision variable and the symmetric matrices $F_i \in \mathbb{R}^{n \times n}$ are known (Boyd et al. 1994).

A.2.6 Input-to-State Stability (ISS)

A system with state x and input u is ISS if and only if it admits a smooth Lyapunov function $V(x)$ such that its variation satisfies:

$$\Delta V(x, u) \leq -\alpha(\|x\|) + \gamma(\|u\|) \tag{A.17}$$

where $\alpha, \gamma \in \mathcal{K}_\infty$.

See (Sontag 2008) for more details. In this manuscript, we use quadratic $\mathcal{K}_\infty$ functions defined as:

$$\alpha(\|x\|) = \alpha x^T M x, \quad \alpha > 0, \ M > 0 \tag{A.18}$$

A.3 Technical Results in Linear Algebra

A.3.1 Congruent Matrices

Two matrices M and N are said to be congruent if there exists an invertible matrix T such that $M = T^T N T$.

Consider two square matrices P and Q, if Q is full rank then P is positive definite if and only if QPQ^T is definite positive.

A.3.2 Schur Complement

Consider a matrix M:

$$M = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \tag{A.19}$$

where A, B, C and D are matrices of appropriate dimensions. The following statements are equivalent:

- $M > 0$
- $A > 0$ and $D - CA^{-1}B > 0$
- $D > 0$ and $A - BD^{-1}C > 0$

Bibliography

Abidi, H., Gerboni, G., Brancadoro, M., Fras, J., Diodato, A., Cianchetti, M., Wurdemann, H., Althoefer, K. & Menciassi, A. (2018), 'Highly dexterous 2-module soft robot for intra-organ navigation in minimally invasive surgery', *The International Journal of Medical Robotics and Computer Assisted Surgery* **14**(1), e1875. 6

Aguiar, B., Berdjag, D., Guerra, T.-M. & Demaya, B. (2018), 'Improving train position accuracy in case of wheel jamming faults', *IFAC-PapersOnLine* **51**(26), 25–30. 24

Allard, J., Cotin, S., Faure, F., Bensoussan, P.-J., Poyer, F., Duriez, C., Delingette, H. & Grisoni, L. (2007), Sofa-an open source framework for medical simulation, *in* 'MMVR 15-Medicine Meets Virtual Reality', Vol. 125, IOP Press, pp. 13–18. 15, 29

Allard, J., Courtecuisse, H. & Faure, F. (2012), Implicit fem solver on gpu for interactive deformation simulation, *in* 'GPU computing gems Jade Edition', Elsevier, pp. 281–294. 27

An, B., Miyashita, S., Ong, A., Tolley, M., Demaine, M., Demaine, E., Wood, R. & Rus, D. (2018), 'An end-to-end approach to self-folding origami structures', *IEEE Transactions on Robotics* **34**(6), 1409–1424. 12

Antoulas, A. C. (2005), *Approximation of large-scale dynamical systems*, Vol. 6, Siam. 55, 60

Apkarian, P. & Tuan, H. D. (2000), 'Parameterized lmis in control theory', *SIAM journal on control and optimization* **38**(4), 1241–1264. 104

Astolfi, A. (2010), 'Model reduction by moment matching for linear and nonlin-

ear systems', *IEEE Transactions on Automatic Control* **55**(10), 2321–2336. 8, 55

Ballantyne, G. H. (2002), 'Robotic surgery, telerobotic surgery, telepresence, and telementoring', *Surgical endoscopy* **16**(10), 1389–1402. 6

Barrachina, S., Benner, P., Quintana-Ortí, E. S. & Quintana-Ortí, G. (2005), Parallel algorithms for balanced truncation of large-scale unstable systems, *in* 'Proceedings of the 44th IEEE Conference on Decision and Control', IEEE, pp. 2248–2253. 60

Benner, P. & Breiten, T. (2012), 'Interpolation-based \calh_2-model reduction of bilinear control systems', *SIAM Journal on Matrix Analysis and Applications* **33**(3), 859–885. 61

Benner, P., Breiten, T. & Damm, T. (2011), 'Generalised tangential interpolation for model reduction of discrete-time mimo bilinear systems', *International Journal of Control* **84**(8), 1398–1407. 61

Benner, P., Cohen, A., Ohlberger, M. & Willcox, K. (2017), *Model Reduction and Approximation: Theory and Algorithms*, Vol. 15, SIAM. 8, 55, 56

Benner, P. & Mitchell, T. (2018), 'Faster and more accurate computation of the h_∞ norm via optimization', *SIAM Journal on Scientific Computing* **40**(5), A3609–A3635. 123

Benner, P., Mitchell, T. & Overton, M. L. (2018), Low-order control design using a reduced-order model with a stability constraint on the full-order model, *in* '2018 IEEE Conference on Decision and Control (CDC)', IEEE, pp. 3000–3005. 74, 123

Benner, P. & Saak, J. (2013), 'Numerical solution of large and sparse continuous time algebraic matrix Riccati and Lyapunov equations: a state of the art survey', *GAMM-Mitteilungen* **36**(1), 32–52. 8

Besselink, B., van de Wouw, N., Scherpen, J. M. & Nijmeijer, H. (2014), 'Model reduction for nonlinear systems by incremental balanced truncation', *IEEE Transactions on Automatic Control* **59**(10), 2739–2753. 60

Boning, P. & Dubowsky, S. (2010), 'Coordinated control of space robot teams for the on-orbit construction of large flexible space structures', *Advanced Robotics* **24**(3), 303–323. 23

Boyd, S., El Ghaoui, L., Feron, E. & Balakrishnan, V. (1994), *Linear matrix inequalities in system and control theory*, Vol. 15, Siam. 24, 108, 124

Brogliato, B. (2016), 'Nonsmooth mechanics: models, dynamics and control'. 111, 115

Brown, E., Rodenberg, N., Amend, J., Mozeika, A., Steltz, E., Zakin, M. R., Lipson, H. & Jaeger, H. M. (2010), 'Universal robotic gripper based on the jamming of granular material', *Proceedings of the National Academy of Sciences* **107**(44), 18809–18814. 7, 8

Bruder, D., Gillespie, B., Remy, C. D. & Vasudevan, R. (2019), 'Modeling and control of soft robots using the koopman operator and model predictive control', *arXiv preprint arXiv:1902.02827* . 22, 23

Calisti, M., Arienti, A., Giannaccini, M. E., Follador, M., Giorelli, M., Cianchetti, M., Mazzolai, B., Laschi, C. & Dario, P. (2010), 'An octopus-bioinspired solution to movement and manipulation for soft robots', *Biomedical Robotics and Biomechatronics (BioRob)* . 5

Calisti, M., Giorelli, M., Levy, G., Mazzolai, B., Hochner, B., Laschi, C. & Dario, P. (2011), 'An octopus-bioinspired solution to movement and manipulation for soft robots', *Bioinspiration & biomimetics* **6**(3), 036002. 12

Chablat, D., Venkateswaran, S. & Boyer, F. (2019), Dynamic model of a bio-inspired robot for piping inspection, *in* 'ROMANSY 22–Robot Design, Dynamics and Control', Springer, pp. 42–51. 6

Chenevier, J., González, D., Aguado, J. V., Chinesta, F. & Cueto, E. (2018), 'Reduced-order modeling of soft robots', *PloS one* **13**(2), e0192052. 17

Cianchetti, M., Ranzani, T., Gerboni, G., Nanayakkara, T., Althoefer, K., Dasgupta, P. & Menciassi, A. (2014), 'Soft robotics technologies to address shortcomings in today's minimally invasive surgery: the stiff-

flop approach', *Soft robotics* **1**(2), 122–131. 6

Coevoet, E., Escande, A. & Duriez, C. (2017), 'Optimization-based inverse model of soft robots with contact handling', *IEEE Robotics and Automation Letters* **2**(3), 1413–1419. 18, 27, 111, 114

Coevoet, E., Escande, A. & Duriez, C. (2019), 'Soft robots locomotion and manipulation control using fem simulation and quadratic programming'. 19

Coevoet, E. et al. (2017), 'Software toolkit for modeling, simulation, and control of soft robots', *Advanced Robotics* **31**(22), 1208–1224. 9, 15, 16, 27, 29, 30, 94, 119

Cook, R. D. et al. (2007), *Concepts and applications of finite element analysis*, John Wiley & Sons. 27

Daafouz, J., Riedinger, P. & Iung, C. (2002), 'Stability analysis and control synthesis for switched systems: a switched lyapunov function approach', *IEEE transactions on automatic control* **47**(11), 1883–1887. 113

Deimel, R. & Brock, O. (2016), 'A novel type of compliant and underactuated robotic hand for dexterous grasping', *The International Journal of Robotics Research* **35**(1-3), 161–185. 5

Della Santina, C., Bianchi, M., Grioli, G., Angelini, F., Catalano, M., Garabini, M. & Bicchi, A. (2017), 'Controlling soft robots: balancing feedback and feedforward elements', *IEEE Robotics & Automation Magazine* **24**(3), 75–83. 21, 101, 118

Della Santina, C., Katzschmann, R. K., Biechi, A. & Rus, D. (2018), Dynamic control of soft robots interacting with the environment, *in* '2018 IEEE International Conference on Soft Robotics (RoboSoft)', IEEE, pp. 46–53. 14, 21, 22, 111

Della Santina, C., Pallottino, L., Rus, D. & Bicchi, A. (2019), 'Exact task execution in highly under-actuated soft limbs: an operational space based approach', *IEEE Robotics and Automation Letters* **4**(3), 2508–2515. 22

Della Santina, C. et al. (2019), 'Learning from humans how to grasp: a data-driven architecture for autonomous grasping with anthropomorphic soft hands', *IEEE Robotics and Automation Letters* **4**(2), 1533–1540. 5

Demourant, F. & Poussot-Vassal, C. (2017), 'A new frequency-domain subspace algorithm with restricted poles location through lmi regions and its application to a wind tunnel test', *International Journal of Control* **90**(4), 779–799. 23

Diao, X. & Ma, O. (2006), Workspace analysis of a 6-dof cable robot for hardware-in-the-loop dynamic simulation, *in* '2006 IEEE/RSJ International Conference on Intelligent Robots and Systems', IEEE, pp. 4103–4108. 18

Ding, Y., Galiana, I., Asbeck, A. T., De Rossi, S. M. M., Bae, J., Santos, T. R. T., de Araújo, V. L., Lee, S., Holt, K. G. & Walsh, C. (2017), 'Biomechanical and physiological evaluation of multi-joint assistance with soft exosuits', *IEEE Transactions on Neural Systems and Rehabilitation Engineering* **25**(2), 119–130. 7

Duriez, C. (2013), Control of elastic soft robots based on real-time finite element method, *in* 'Robotics and Automation (ICRA), 2013 IEEE International Conference on', IEEE, pp. 3982–3987. 18, 48, 49

Falkenhahn, V., Hildebrandt, A., Neumann, R. & Sawodny, O. (2015), Model-based feedforward position control of constant curvature continuum robots using feedback linearization, *in* '2015 IEEE International Conference on Robotics and Automation (ICRA)', IEEE, pp. 762–767. 21

Fang, G., Wang, X., Wang, K., Lee, K.-H., Ho, J. D., Fu, H.-C., Fu, D. K. C. & Kwok, K.-W. (2019), 'Vision-based online learning kinematic control for soft robots using local gaussian process regression', *IEEE Robotics and Automation Letters* **4**(2), 1194–1201. 20

Faubourg, L. & Pomet, J.-B. (2000), 'Control lyapunov functions for homogeneous jurdjevic-quinn systems'. 44, 45

Felt, W., Chin, K. Y. & Remy, C. D. (2017), 'Smart Braid Feedback for the

Closed-Loop Control of Soft Robotic Systems', *Soft Robotics* **4**(3), 261–273. 12

Felton, S., Tolley, M., Demaine, E., Rus, D. & Wood, R. (2014), 'A method for building self-folding machines', *Science* **345**(6197), 644–646. 12

Fries, F., Miyashita, S., Rus, D., Pfeifer, R. & Damian, D. D. (2014), Electromagnetically driven elastic actuator, *in* 'Robotics and Biomimetics (ROBIO), 2014 IEEE International Conference on', IEEE, pp. 309–314. 12

Gahinet, P. & Apkarian, P. (1994), 'A linear matrix inequality approach to h_∞ control', *International journal of robust and nonlinear control* **4**(4), 421–448. 24

Geromel, J. C. & Colaneri, P. (2006*a*), 'Robust stability of time varying polytopic systems', *Systems & Control Letters* **55**(1), 81–85. 104

Geromel, J. C. & Colaneri, P. (2006*b*), 'Stability and stabilization of discrete time switched systems', *International Journal of Control* **79**(07), 719–728. 113, 114

Gillespie, M. T., Best, C. M. & Killpack, M. D. (2016), Simultaneous position and stiffness control for an inflatable soft robot, *in* '2016 IEEE international conference on robotics and automation (ICRA)', IEEE, pp. 1095–1101. 12

Goury, O. & Duriez, C. (2018), 'Fast, generic, and reliable control and simulation of soft robots using model order reduction', *IEEE Transactions on Robotics* pp. 1–12. 17, 53, 55

Greer, J. D., Morimoto, T. K., Okamura, A. M. & Hawkes, E. W. (2019), 'A soft, steerable continuum robot that grows via tip extension', *Soft robotics* **6**(1), 95–108. 5

Gugercin, S., Antoulas, A. C. & Beattie, C. (2008), 'H_2 model reduction for large-scale linear dynamical systems', *SIAM journal on matrix analysis and applications* **30**(2), 609–638. 56, 61

Haber, A. & Verhaegen, M. (2016), 'Sparse solution of the lyapunov equation for large-scale interconnected systems', *Automatica* **73**, 256–268. 8

Hall, J. F. (2006), 'Problems encountered from the use (or misuse) of rayleigh damping', *Earthquake engineering & structural dynamics* **35**(5), 525–545. 33

Hao, Y., Gong, Z., Xie, Z., Guan, S., Yang, X., Ren, Z., Wang, T. & Wen, L. (2016), Universal soft pneumatic robotic gripper with variable effective length, *in* '2016 35th Chinese Control Conference (CCC)', IEEE, pp. 6109–6114. 7

Hauser, H., Ijspeert, A. J., Füchslin, R. M., Pfeifer, R. & Maass, W. (2012), 'The role of feedback in morphological computation with compliant bodies', *Biological cybernetics* **106**(10), 595–613. 22

Hughes, J., Scimeca, L., Ifrim, I., Maiolino, P. & Iida, F. (2018), 'Achieving robotically peeled lettuce', *IEEE Robotics and Automation Letters* **3**(4), 4337–4342. 6

Ichalal, D., Marx, B., Ragot, J. & Maquin, D. (2009), Simultaneous state and unknown inputs estimation with PI and PMI observers for Takagi Sugeno model with unmeasurable premise variables, *in* 'Control and Automation, 2009. MED'09. 17th Mediterranean Conference on', IEEE, pp. 353–358. 67, 68

Jamone, L., Natale, L., Sandini, G. & Takanishi, A. (2012), Interactive online learning of the kinematic workspace of a humanoid robot, *in* '2012 IEEE/RSJ International Conference on Intelligent Robots and Systems', IEEE, pp. 2606–2612. 18

Jurdjevic, V. & Quinn, J. P. (1978), 'Controllability and stability', *Journal of differential equations* **28**(3), 381–389. 44

Katzschmann, R. K., Della Santina, C., Toshimitsu, Y., Bicchi, A. & Rus, D. (2019), 'Dynamic motion control of multi-segment soft robots using piecewise constant curvature matched with an augmented rigid body model'. 13, 14, 22

Katzschmann, R. K., Marchese, A. D. & Rus, D. (2016), Hydraulic autonomous soft robotic fish for 3d swimming, *in* 'Experimental Robotics', Springer, pp. 405–420. 5, 12

Katzschmann, R., Thieffry, M. et al. (2019), Dynamically closed-loop controlled soft robotic arm using a reduced order finite element model with state observer, *in* 'IEEE 2019 International Conference on Soft Robotics'. 9, 23, 74, 82, 119

Kim, S., Laschi, C. & Trimmer, B. (2013), 'Soft robotics: a bioinspired evolution in robotics', *Trends in biotechnology* **31**(5), 287–294. 4, 12

King, C.-H., Culjat, M. O., Franco, M. L., Bisley, J. W., Dutson, E. & Grundfest, W. S. (2008), 'Optimization of a pneumatic balloon tactile display for robot-assisted surgery based on human perception', *IEEE Transactions on Biomedical Engineering* **55**(11), 2593–2600. 6

Kruszewski, A., Wang, R. & Guerra, T.-M. (2008), 'Nonquadratic stabilization conditions for a class of uncertain nonlinear discrete time ts fuzzy models: A new approach', *IEEE Transactions on Automatic Control* **53**(2), 606–611. 108

Lall, S., Marsden, J. E. & Glavaški, S. (2002), 'A subspace approach to balanced truncation for model reduction of nonlinear control systems', *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal* **12**(6), 519–535. 60

Lamnabhi-Lagarrigue, F., Annaswamy, A., Engell, S., Isaksson, A., Khargonekar, P., Murray, R. M., Nijmeijer, H., Samad, T., Tilbury, D. & den Hof, P. V. (2017), 'Systems and Control for the future of humanity, research agenda: Current and future roles, impact and grand challenges', *Annual Reviews in Control* **43**, 1–64.
URL: *http://www.sciencedirect.com/science/article/pii/S1367578817300573* 5

Lanfranco, A. R., Castellanos, A. E., Desai, J. P. & Meyers, W. C. (2004), 'Robotic surgery: a current perspective', *Annals of surgery* **239**(1), 14. 6

Laumond, J.-P. (2012), 'Robotics: Hephaestus does it again'. 3

Liberzon, D. (2003), *Switching in systems and control*, Springer Science & Business Media. 113

Lismonde, A., Sonneville, V. & Brüls, O. (2017), 'Trajectory planning of soft link robots with improved intrinsic safety', *IFAC-PapersOnLine* **50**(1), 6016–6021. 18

Löfberg, J. (2004), Yalmip : A toolbox for modeling and optimization in matlab, *in* 'In Proceedings of the CACSD Conference', Taipei, Taiwan. 79

Majidi, C. (2014), 'Soft robotics: a perspective — current trends and prospects for the future', *Soft Robotics* **1**(1), 5–11. 4

Marchese, A. D., Katzschmann, R. K. & Rus, D. (2015), 'A recipe for soft fluidic elastomer robots', *Soft Robotics* **2**(1), 7–25. 37

Marchese, A. D., Komorowski, K., Onal, C. D. & Rus, D. (2014), Design and control of a soft and continuously deformable 2d robotic manipulation system, *in* '2014 IEEE international conference on robotics and automation (ICRA)', IEEE, pp. 2189–2196. 13

Marchese, A. D. & Rus, D. (2016), 'Design, kinematics, and control of a soft spatial fluidic elastomer manipulator', *International Journal of Robotics Research* **35**(7), 840–869. 37

Marchese, A. D., Tedrake, R. & Rus, D. (2016), 'Dynamics and trajectory optimization for a soft spatial fluidic elastomer manipulator', *The International Journal of Robotics Research* **35**(8), 1000–1019. 18

McMahan, W., Chitrakaran, V., Csencsits, M., Dawson, D., Walker, I. D., Jones, B. A., Pritts, M., Dienno, D., Grissom, M. & Rahn, C. D. (2006), Field trials and testing of the octarm continuum manipulator, *in* 'Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.', IEEE, pp. 2336–2341. 5

Megretski, A. (2006), H-infinity model reduction with guaranteed suboptimality

bound, *in* '2006 American Control Conference', IEEE, pp. 6–pp. 123

Mohammad, D., Khan, N. & Ramamurti, V. (1995), 'On the role of rayleigh damping', *Journal of Sound and Vibration* **185**(2), 207–218. 33

Morales Bieze, T. (2017), Contribution to the kinematic modeling and control of soft manipulators using computational mechanics, PhD thesis, Lille 1. 4, 114, 115

Morales Bieze, T., Largilliere, F., Kruszewski, A., Zhang, Z., Merzouki, R. & Duriez, C. (2018), 'Finite element method-based kinematics and closed-loop control of soft, continuum manipulators', *Soft robotics* **5**(3), 348–364. 114

Morzadec, T., Marcha, D. & Duriez, C. (2019), Toward shape optimization of soft robots, *in* '2019 2nd IEEE International Conference on Soft Robotics (RoboSoft)', IEEE, pp. 521–526. 118

Nakajima, K., Li, T., Hauser, H. & Pfeifer, R. (2014), 'Exploiting short-term memory in soft body dynamics as a computational resource', *Journal of The Royal Society Interface* **11**(100), 20140437. 23

Nguyen, A.-T., Guerra, T.-M. & Sentouh, C. (2018), 'Simultaneous estimation of vehicle lateral dynamics and driver torque using lpv unknown input observer', *IFAC-PapersOnLine* **51**(26), 13–18. 24

Nguyen, V.-A., Nguyen, A.-T., Dequidt, A., Vermeiren, L. & Dambrine, M. (2019), 'Nonlinear tracking control with reduced complexity of serial robots: A robust fuzzy descriptor approach', *International Journal of Fuzzy Systems* pp. 1–13. 24

Niiyama, R., Sun, X., Sung, C., An, B., Rus, D. & Kim, S. (2015), 'Pouch motors: printable soft actuators integrated with computational design', *Soft Robotics* **2**(2), 59–70. 12

Onal, C. D. & Rus, D. (2013), 'Autonomous undulatory serpentine locomotion utilizing body dynamics of a fluidic soft robot', *Bioinspiration & biomimetics* **8**(2), 026003. 18

Onal, C. D., Wood, R. J. & Rus, D. (2013), 'An origami-inspired approach to worm robots', *IEEE/ASME Transactions on Mechatronics* **18**(2), 430–438. 12

O'Neill, C. T., Phipps, N. S., Cappello, L., Paganoni, S. & Walsh, C. J. (2017), A soft wearable robot for the shoulder: Design, characterization, and preliminary testing, *in* 'Rehabilitation Robotics (ICORR), 2017 International Conference on', IEEE, pp. 1672–1678. 7

Payne, C. J., Wamala, I., Abah, C., Thalhofer, T., Saeed, M., Bautista-Salinas, D., Horvath, M. A., Vasilyev, N. V., Roche, E. T., Pigula, F. A. & Others (2017), 'An implantable extracardiac soft robotic device for the failing heart: Mechanical coupling and synchronization', *Soft robotics* **4**(3), 241–250. 7

Petit, N. & Rouchon, P. (2001), 'Flatness of heavy chain systems', *SIAM Journal on Control and Optimization* **40**(2), 475–495. 24

Polygerinos, P., Wang, Z., Galloway, K. C., Wood, R. J. & Walsh, C. J. (2015), 'Soft robotic glove for combined assistance and at-home rehabilitation', *Robotics and Autonomous Systems* **73**, 135–143. 7

Poussot-Vassal, C. & Vuillemin, P. (2012), 'Introduction to {MORE}: a {MO}del {RE}duction Toolbox', *2012 IEEE International Conference on Control Applications (CCA) Part of 2012 IEEE Multi-Conference on Systems and Control October 3-5, 2012. Dubrovnik, Croatia* . 53, 65, 95

Reddy, J. N. (1993), 'An introduction to the finite element method', *New York* . 27

Renda, F., Boyer, F., Dias, J. & Seneviratne, L. (2017), 'Discrete cosserat approach for multi-section soft robots dynamics', *arXiv preprint arXiv:1702.03660* . 14

Renda, F., Cacucciolo, V., Dias, J. & Seneviratne, L. (2016), Discrete cosserat approach for soft robot dynamics: A new piece-wise constant strain model with torsion and shears, *in* '2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)', IEEE, pp. 5495–5502.

15

Rowley, C. W. (2005), 'Model reduction for fluids, using balanced proper orthogonal decomposition', *International Journal of Bifurcation and Chaos* **15**(03), 997–1013. 60

Rozen-Levy, S., Messner, W. & Trimmer, B. A. (2019), 'The design and development of branch bot: a branch-crawling, caterpillar-inspired, soft robot', *The International Journal of Robotics Research* p. 0278364919846358. 19

Ryan, T. & Kim, H. J. (2013), 'Lmi-based gain synthesis for simple robust quadrotor control', *IEEE Transactions on Automation Science and Engineering* **10**(4), 1173–1178. 24

Sadati, S. H., Shiva, A., Renson, L., Rucker, C., Althoefer, K., Nanayakkara, T., Bergeles, C., Hauser, H. & Walker, I. D. (2019), 'Reduced order vs. discretized lumped system models with absolute and relative states for continuum manipulators'. 17

Sadati, S. M. H., Naghibi, S. E., Walker, I. D., Althoefer, K. & Nanayakkara, T. (2018), 'Control Space Reduction and Real-Time Accurate Modeling of Continuum Manipulators Using Ritz and Ritz–Galerkin Methods', *IEEE Robotics and Automation Letters* . 22

Scimeca, L., Daniel, C.-C., Angel, P., Antonio, M., Iida, F. et al. (2019), 'Non-destructive robotic assessment of mango ripeness via multi-point soft haptics'. 6

Shepherd, R. F., Ilievski, F., Choi, W., Morin, S. A., Stokes, A. A., Mazzeo, A. D., Chen, X., Wang, M. & Whitesides, G. M. (2011), 'Multigait soft robot', *Proceedings of the national academy of sciences* **108**(51), 20400–20403. 5

Slade, P., Gruebele, A., Hammond, Z., Raitor, M., Okamura, A. M. & Hawkes, E. W. (2017), Design of a soft catheter for low-force and constrained surgery, *in* '2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)', IEEE, pp. 174–180. 6

Sontag, E. D. (2008), Input to state stability: Basic concepts and results, *in* 'Nonlinear and optimal control theory', Springer, pp. 163–220. 69, 124

Soter, G., Garrad, M., Conn, A. T., Hauser, H. & Rossiter, J. (2019), Skinflow: A soft robotic skin based on fluidic transmission, *in* '2019 2nd IEEE International Conference on Soft Robotics (RoboSoft)', IEEE, pp. 355–360. 13

Stürzer, D., Arnold, A. & Kugi, A. (2018), 'Closed-loop stability analysis of a gantry crane with heavy chain and payload', *International Journal of Control* **91**(8), 1931–1943. 23

Taylor, R., Menciassi, A., Fichtinger, G., Fiorini, P. & Dario, P. (2016), *Medical Robotics and Computer-Integrated Surgery.*, Springer Handbook of Robotics. Springer. 6

Thieffry, M., Kruszewski, A., Duriez, C. & Guerra, T.-M. (2019), 'Control design for soft robots based on reduced-order model', *IEEE Robotics and Automation Letters* **4**(1), 25–32. 9, 23, 44, 53, 67, 74, 82, 119

Thieffry, M., Kruszewski, A., Goury, O., Guerra, T.-M. & Duriez, C. (2017), Dynamic control of soft robots, *in* 'IFAC World congress'. 23, 119

Thieffry, M., Kruszewski, A., Guerra, T.-M. & Duriez, C. (2018), Reduced Order Control of Soft Robots with Guaranteed Stability, *in* 'European Control Conference ECC18', Limassol, Cyprus. 9, 44, 47, 48, 53, 79, 119

Thuruthel, T. G. et al. (2018*a*), 'Control strategies for soft robotic manipulators: A survey', *Soft robotics* . 18, 21

Thuruthel, T. G. et al. (2018*b*), 'Stable open loop control of soft robotic manipulators', *IEEE Robotics and Automation Letters* **3**(2), 1292–1298. 18

Tolley, M. T., Shepherd, R. F., Mosadegh, B., Galloway, K. C., Wehner, M., Karpelson, M., Wood, R. J. & Whitesides, G. M. (2014), 'A resilient, untethered soft robot', *Soft robotics* **1**(3), 213–223. 6

Tóth, R. (2010), *Modeling and identification of linear parameter-varying systems*, Vol. 403, Springer. 104

Trivedi, D., Lotfi, A. & Rahn, C. D. (2008), 'Geometrically exact models for soft robotic manipulators', *IEEE Transactions on Robotics* **24**(4), 773–780. 14

Trivedi, D. et al. (2008), 'Soft robotics: Biological inspiration, state of the art, and future research', *Applied bionics and biomechanics* **5**(3), 99–117. 11

Truby, R. L., Katzschmann, R. K., Lewis, J. A. & Rus, D. (2019), Soft robotic fingers with embedded ionogel sensors and discrete actuation modes for somatosensitive manipulation, *in* '2019 2nd IEEE International Conference on Soft Robotics (RoboSoft)', IEEE, pp. 322–329. 12, 13

Umedachi, T., Vikas, V. & Trimmer, B. A. (2013), Highly deformable 3-d printed soft robot generating inching and crawling locomotions with variable friction legs, *in* '2013 IEEE/RSJ international conference on Intelligent Robots and Systems', IEEE, pp. 4590–4595. 18

Van Dooren, P., Gallivan, K. A. & Absil, P.-A. (2008), 'H2-optimal model reduction of mimo systems', *Applied Mathematics Letters* **21**(12), 1267–1273. 56

Vieira, A. (2018), Optimal control of linear complementarity systems, PhD thesis, Université Grenoble Alpes. 112

Vigne, M., El Khoury, A., Masselin, M., Di Meglio, F. & Petit, N. (2018), Estimation of multiple flexibilities of an articulated system using inertial measurements, *in* '2018 IEEE Conference on Decision and Control (CDC)', IEEE, pp. 6779–6785. 7

Vuillemin, P. (2014), Frequency-limited model approximation of large-scale dynamical models, PhD thesis, ISAE-Institut Supérieur de l'Aéronautique et de l'Espace. 56, 60

Vuillemin, P., Poussot-Vassal, C. & Alazard, D. (2014), 'Two upper bounds on the h_∞-norm of lti dynamical systems', *IFAC Proceedings Volumes*

47(3), 5562–5567. 123

Wang, H., Yang, B., Liu, Y., Chen, W., Liang, X. & Pfeifer, R. (2017), 'Visual servoing of soft robot manipulator in constrained environments with an adaptive controller', *IEEE/ASME Transactions on Mechatronics* **22**(1), 41–50. 20

Wang, T. & Liu, B. (2016), Different polytopic decomposition for visual servoing system with lmi-based predictive control, *in* '2016 35th Chinese Control Conference (CCC)', IEEE, pp. 10320–10324. 24

Wang, W. & Ahn, S.-H. (2017), 'Shape memory alloy-based soft gripper with variable stiffness for compliant and effective grasping', *Soft robotics* **4**(4), 379–389. 12

Webster, R. J. & Jones, B. A. (2010), 'Design and kinematic modeling of constant curvature continuum robots: A review', *The International Journal of Robotics Research* **29**(13), 1661–1683. 13

Yip, M. C. & Camarillo, D. B. (2016), 'Model-less hybrid position/force control: a minimalist approach for continuum manipulators in unknown, constrained environments', *IEEE Robotics and Automation Letters* **1**(2), 844–851. 20

You, X., Zhang, Y., Chen, X., Liu, X., Wang, Z., Jiang, H. & Chen, X. (2017), Model-free control for soft manipulators based on reinforcement learning, *in* '2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)', IEEE, pp. 2909–2915. 20

Zhang, S., He, W. & Huang, D. (2016), 'Active vibration control for a flexible string system with input backlash', *IET Control Theory & Applications* **10**(7), 800–805. 23

Zhang, Z., Morales Bieze, T., Dequidt, J., Kruszewski, A. & Duriez, C. (2017), Visual Servoing Control of Soft Robots based on Finite Element Model, *in* 'IROS 2017 - IEEE/RSJ International Conference on Intelligent Robots and Systems', Vancouver, Canada.
URL: *https://hal.archives-ouvertes.fr/hal-01618330* 21

Zheng, G., Goury, O., Thieffry, M., Kruszewski, A. & Duriez, C. (2019), Controllability pre-verification of silicone soft robots based on finite-element method, *in* '2019 IEEE International Conference on Robotics and Automation (ICRA)', IEEE. 118

Zhou, K., Salomon, G. & Wu, E. (1999), 'Balanced realization and model reduction for unstable systems', *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal* **9**(3), 183–198. 60