Vision-based Calibration, Position Control and Force Sensing for Soft Robots

Calibration basée sur la vision, contrôle de position et détection de force pour robots doux

Zhongkai Zhang

Thèse préparée à L'INRIA et CRIStAL,
soutenue publiquement à l'Université de Lille,
le 10/01/2019,
pour obtenir le grade de Docteur en Informatique

Membres du jury :

Directeur:	Christian Duriez	Directeur de recherche, INRIA
	Jeremie Dequidt	Maître de conférences, Université de Lille
Rapporteur:	Robert Howe	Professeur, Harvard University
	Danail Stoyanov	Professeur, University College London
Examinateur:	Jessica Burgner-Kahrs	Professeur associé, Leibniz Universität Hannover
	Hongbin Liu	Professeur associé, King's College London
Président:	Cecilia Laschi	Professeur, Scuola Superiore Sant'Anna, Pisa

Acknowledgements

First of all, I would like to thank my advisors, Jeremie Dequidt and Christian Duriez. Thank you for giving me the opportunity to join your team. During the three years, your discussion and guidance are very important for me to broaden my research interest and to make this scientific work possible.

I would like to thank INRIA and the Region Hauts de France for the finance of my research.

I would like to show my acknowledgment to my jury members: Prof. Robert Howe, Prof. Danail Stoyanov, Prof. Cecilia Laschi, Prof. Jessica Burgner-Kahrs, and Prof. Hongbin Liu. Thank you for the review of my thesis. Your comments are really valuable for the improvement of this work.

I would like to thank also Prof. Alexandre Kruszewski (INRIA) for the help of automatic control, Dr. Antoine Petit (INRIA) for the help of image processing, Dr. Junghwan Back (KCL) for the help of experiments on the catheter robot.

Many thanks to all members in DEFROST team. I am grateful for the time spent with them.

I would like to express my gratitude to my parents, and my wife for their love and encouragement.

Abstract

Soft robots can interact with the environment in a safe and compliant way because of their deformable structures. However, the modeling of soft robots which have, theoretically, infinite degrees of freedom, are extremely difficult especially when the robots have complex configurations. This difficulty of modeling leads to new challenges for the calibration and the control design of the robots, but also new opportunities with possible new force sensing strategies. This dissertation aims to provide new and general solutions using modeling and vision.

The thesis at first presents a discrete-time kinematic model for soft robots based on the real-time Finite Element (FE) method. Then, a vision-based simultaneous calibration of sensor-robot system and actuators is investigated. Two closed-loop position controllers are designed and the robust stability of the closed-loop system is analyzed using Lyapunov stability theory. Besides, to deal with the problem of image feature loss, a switched control strategy is proposed by combining both the open-loop controller and the closed-loop controller.

Using soft robot itself as a force sensor is available due to the deformable feature of soft structures. Two methods (marker-based and marker-free) of external force sensing for soft robots are proposed based on the fusion of vision-based measurements and FE model. Using both methods, not only the intensities but also the locations of the external forces can be estimated. The marker-based approach is proposed to find the correct locations of external forces among several possible ones. If there are no obvious feature points on the surface of the soft robot, the marker-free force sensing strategy is available using an RGB-D camera.

As a specific application, a cable-driven continuum catheter robot through contacts is modeled based on FE method. Then, the robot is controlled by a decoupled control strategy which allows to control insertion and bending independently. Both the control inputs and the contact forces along the entire catheter can be computed by solving a quadratic programming (QP) problem with a linear complementarity constraint (QPCC). A simplified solution is proposed for the computation of QPCC by converting it into a standard QP problem.

Keywords: Soft robots, catheter robot, calibration, closed-loop control, external force sensing, contact, registration, RGB-D camera, Finite Element Method

Résumé

La modélisation de robots souples est extrêmement difficile, à cause notamment du nombre théoriquement infini des degrés de liberté. Cette difficulté est accentuée lorsque les robots ont des configurations complexes. Ce problème de modélisation entraîne de nouveaux défis pour la calibration et la conception des commandes des robots, mais également de nouvelles opportunités avec de nouvelles stratégies de détection de force possibles. Cette thèse a pour objectif de proposer des solutions nouvelles et générales utilisant la modélisation et la vision.

La thèse présente dans un premier temps un modèle cinématique à temps discret pour les robots souples reposant sur la méthode des éléments finis (FEM) en temps réel. Ensuite, une méthode de calibration basée sur la vision du système de capteur-robot et des actionneurs est étudiée. Deux contrôleurs de position en boucle fermée sont conçus. En outre, pour traiter le problème de la perte d'image, une stratégie de commande commutable est proposée en combinant à la fois le contrôleur à boucle ouverte et le contrôleur à boucle fermée.

Deux méthodes (avec et sans marqueur(s)) de détection de force externe pour les robots déformables sont proposées. L'approche est basée sur la fusion de mesures basées sur la vision et le modèle par FEM. En utilisant les deux méthodes, il est possible d'estimer non seulement les intensités, mais également l'emplacement des forces externes.

Enfin, nous proposons une application concrète: un robot cathéter dont la flexion à l'extrémité est piloté par des câbles. Le robot est contrôlé par une stratégie de contrôle découplée qui per-met de contrôler l'insertion et la flexion indépendamment, tout en se basant sur un modèle FEM.

Mots clés: Robots doux, robot de cathéter, étalonnage, contrôle en boucle fermée, détection de force externe, contact, Caméra RGB-D, Méthode des éléments finis

NOMENCLATURE

$\mathbf{x}$ Position vector of all FEM nodes.
$\hat{\mathbf{x}}$ Estimated position vector of all FEM nodes.
$d\mathbf{x}$ Incremental displacement of nodes.
$\mathbf{f}_{ext}$ External load vector.
$\mathbf{f}(\mathbf{x})$ Internal stiffness forces vector.
$\mathbf{K}(\mathbf{x})$ Tangent stiffness matrix.
$\mathbf{H}_a$ Mapping matrix between nodes and actuator directions.
$\mathbf{H}_e$ Mapping matrix between nodes and effector directions.
$\mathbf{H}_f$ Mapping matrix between nodes and external forces directions.
$\mathbf{H}_c$ Mapping matrix between nodes and contact directions.
λ_a Force contribution vector of actuators.
λ_f Force contribution vector of external forces.
λ_c Force contribution vector of contacts.
λ_I Force contribution of actuators and external forces.
λ Force contribution of actuators and contacts.
δ_a Position vector in the actuator space.
δ_e Position vector of effectors.
δ_c Position vector in the contact space.
$\mathbf{W}_{ea}(\mathbf{x})$ Compliance matrix between effectors and actuators.
$\mathbf{W}_{ef}(\mathbf{x})$ Compliance matrix between effectors and external forces.
$\mathbf{W}_{ec}(\mathbf{x})$ Compliance matrix between effectors and contacts.
$\mathbf{W}_{aa}(\mathbf{x})$ Compliance matrix between actuators.
$\mathbf{W}_{af}(\mathbf{x})$ Compliance matrix between actuators and external forces.
$\mathbf{W}_{ac}(\mathbf{x})$ Compliance matrix between actuators and contacts.
$\mathbf{W}_{ca}(\mathbf{x})$ Compliance matrix between contacts and actuators.
$\mathbf{W}_{eI}(\mathbf{x})$ Compliance matrix between effectors and actuators, external forces.
$\mathbf{W}_{e}(\mathbf{x})$ Compliance matrix between effectors and actuators, contacts.
$\mathbf{W}_{a}(\mathbf{x})$ Compliance matrix between actuators and actuators, contacts.
$\mathbf{W}_{c}(\mathbf{x})$ Compliance matrix between contacts and actuators, contacts.
δ_e^{free} Position vector of effectors when $\lambda_a = 0$ and $\lambda_c = 0$.
δ_a^{free} Position vector of actuators when $\lambda_a = 0$ and $\lambda_c = 0$.
δ_c^{free} Position vector of contacts when $\lambda_a = 0$ and $\lambda_c = 0$.
$\triangle\lambda_a$ Incremental force contribution of actuators.
$\triangle\lambda_f$ Incremental force contribution of external forces.
$\triangle\lambda_I$ Incremental force contribution of actuators and external forces.

$\triangle\delta_e$ Incremental position of effectors.

$\triangle\delta_a$ Incremental position contribution of actuators.

$\triangle\delta_f$ Incremental position contribution of external forces.

$\mathbf{J}(\mathbf{x})$ Jacobian Matrix for soft robot.

$\hat{\mathbf{J}}(\hat{\mathbf{x}})$ The predicted Jacobian matrix.

$V(\mathbf{x})$ Lyapunov candidate function.

$\mathbf{R}$ Rotation matrix for rigid transformation.

$\mathbf{T}$ Translation vector for rigid transformation.

δ_{robot} Position vector of feature points in the robot frame.

δ_{cam} Position vector of feature points in the camera frame.

δ_{tran} Transformed δ_{cam} using optimal rigid transformation.

$\delta_{e,d}$ Desired position of effector.

$\mathbf{e}$ Position error of effector $\mathbf{e} = \delta_e - \delta_{e,d}$.

Γ The objective function.

Ω Constraint for actuator inputs.

$\mathbf{w}_i$ The position of the particle i for the particle swarm optimization (PSO)

$\mathbf{s}_i$ The velocity of the particle i for PSO.

c_1, c_2 Two acceleration constants for PSO.

p_i The best solution which has achieved so far by particle i for PSO.

g_i The global best solution obtained by the swarm for PSO.

R_1, R_2 Two random numbers between 0 and 1 for PSO.

$\omega(t)$ The inertia weight for PSO.

ω_{start} The initial value of the inertia weight.

ω_{end} The final value of the inertia weight.

t_{max} The maximum number of allowable iterations for PSO.

t The maximum number of the current iteration times for PSO.

I_1 The maximum iteration for the optimization of rigid transformation.

I_2 The maximum iteration for the optimization of soft deformation.

μ Control parameter for the first implementation of closed-loop control.

k_p Control parameter for the second implementation of closed-loop control.

γ A scalar to show the degree of difference between real robot and its FE model.

α The accessorial parameter for the proof of robust stability.

p A constant parameter to tune the length of the switching time period.

$*^s$ Corresponding variables $*$ for the FE model.

λ_f^r The external forces of soft robot for the force sensing.

δ_e^r The position effectors of soft robot for the force sensing.

λ_a^r The actuator force of soft robot for the force sensing.
δ_a^r The actuator displacement of soft robot for the force sensing.
d_m A threshold distance to discard potential outlier feature point.
d_l A threshold to discard registrated points from wrong segmented point cloud.
r_{search} Constant search radius using KdTree.
n_{neigh} A threshold number of neighbors for the feature points detection.
d_{mini} The minimum distance between the effector and the normal line.
δ_e^N The mapped value of δ_e on the normal line of the surface.
$\delta_e^{free,N}$ The mapped value of δ_e^{free} on the normal line of the surface.
$\delta_a^{free,N}$ The mapped value of δ_a^{free} on the normal line of the surface.
$\mathbf{W}_{ea}^N$ The mapped value of $\mathbf{W}_{ea}$ on the normal line of the surface.
$\mathbf{W}_{ef}^N$ The mapped value of $\mathbf{W}_{ef}$ on the normal line of the surface.
$\mathbf{W}_{aa}^N$ The mapped value of $\mathbf{W}_{aa}$ on the normal line of the surface.
$\mathbf{W}_{af}^N$ The mapped value of $\mathbf{W}_{af}$ on the normal line of the surface.
$\mathbf{W}_{eI}^N$ The mapped value of $\mathbf{W}_{eI}$ on the normal line of the surface.
δ_d^N The position of the registered points along the normal direction.
I_e Estimation error of external forces.
I_w Evaluation index for the distribution of markers and candidate locations.
P The nearest point on the trajectory to the catheter tip.
Px The tangent vector for the trajectory at the point P.
Py The projection vector at the point P.
Pz The normal vector at the point P.
δ_e^{3D} 3D position of catheter tip.
$\delta_e^{3D}P$ The 3D position error of catheter tip.
ε The parameter to address the linear complementarity constraint.
K_P The proportional gain for the control of catheter robot.
K_D The differential gain for the control of catheter robot.

Contents

Chapter 1

Introduction

1.1 Introduction of Soft Robot

1.1.1 Bio-inspiration and Applications

Biological systems provide many examples of soft structures which are able to bend, extend and twist [1]. Some have a completely soft body, such as caterpillar and octopus, and at the same time exhibits sophisticated behaviors; some have rigid bones with membrane or skin covering them and joining them together, like bat and fish. Obviously, the soft structures are essential for their unrivaled agility and maneuvering characteristics.

Inspired by nature, engineers have developed many robots which have a continuously deformable structure (see Fig. 1.1). Softworms [2] are inspired by studies on the locomotion of caterpillars and are capable of crawling and steering. The Softworms are electrically powered using shape-memory alloy coils and motor-tendons. Inspired by the octopus, several robots have been developed [3, 4] to mimic octopus crawling locomotion. Robot fish realizes the biomimetic swimming with a fully integrated on-board system for power and remote control [5, 6]. The robot Bat Bot is a fully self-contained, autonomous flying robot that mimic the morphological properties of bat wings. The highly stretchable silicone-based membrane wings provide safety for humans and the robot itself [7].

In this thesis, soft robots are defined as robots which create motion by deforming their structures. Soft robots can be made of soft materials which are similar to soft biological materials [8]. In this case, they are also named as soft material robots. Soft robots can also be made of rigid materials but have a low structural stiffness, like some continuum robots we can find the literature [9]. Both soft material robots and continuum robots have similar features so that the modeling and control methods are similar, especially for manipulators. The work presented in this thesis will also applied to them.

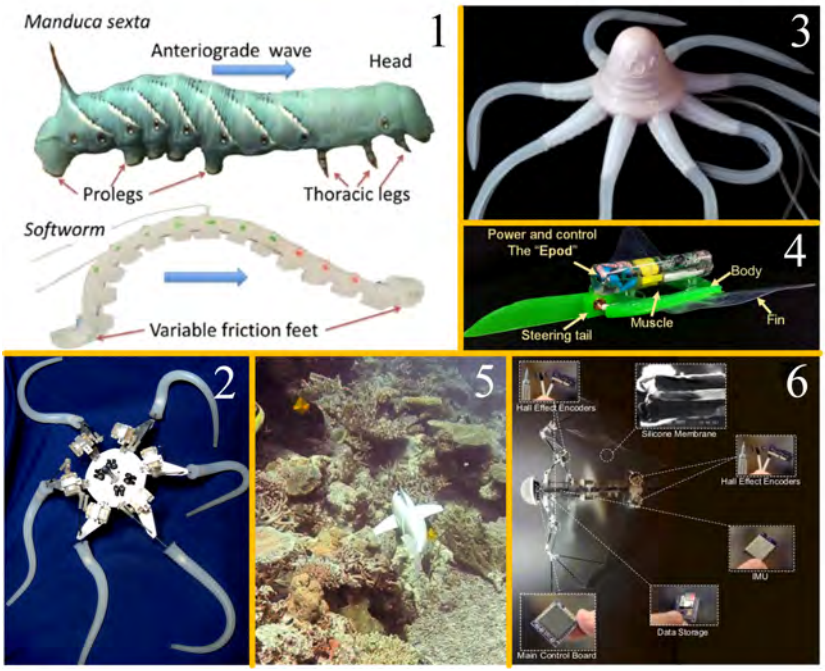

Figure 1.1: Bio-inspired design of soft robots: 1. softworms [2]; 2 and 3. octopus robots [3, 4]; 4 and 5. Robot fish [5, 6]; 6. Bat Bot [7].

Due to their low stiffness of structure, soft robots can reduce the harm to interact with humans and show more adaptation to the environment. As is shown in Fig. 1.2, the advantages of soft robots make them more suitable to be used in confined spaces [10], for manipulation of objects with different shapes [11] and for medical applications [12–14].

1.1.2 Actuators

The promising applications stimulate researchers to develop effective soft actuation technologies for soft robots [15]. Soft actuators are employed to generate deformation for soft robots. Some actuators are made of advanced materials like shape memory alloys and dielectric elastomers. Some other actuators are made of traditional materials but with special structures. There are various methods for the actuation of soft robots (see Fig. 1.3), like fluidic elastomer actuators [16], fiber reinforced soft bending actuator [17], pneumatic artificial muscles [18], shape memory alloy actuators [19], dielectric elastomer actuators [20] and cable actuators [21].

Enlarging the choice of power resources has also been investigated in soft robotics (see Fig. 1.4). The popular resources can be separated as voltage [20], vacuum [22], air pressure [23], chemical fuel [24], motor [21] and magnetic field [25].

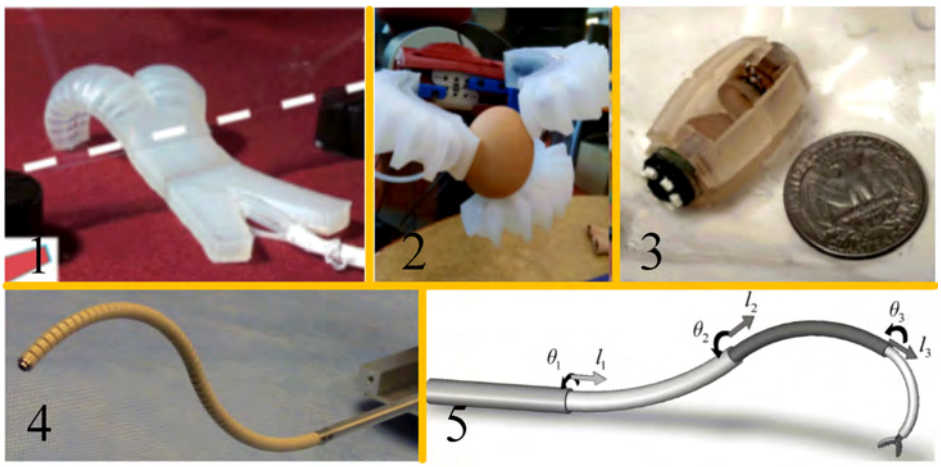

Figure 1.2: Applications of soft robots: 1. movement in confined space [10]; 2. manipulation of objects [11]; 3, 4 and 5. medical applications [12–14].

1.1.3 Soft Sensors

In soft robotics, there is a need to have sensors that are compatible with the large compliance of the robots. In addition, the community is looking to solution with low fabrication cost, light weight and high impact resistance. Usually, soft sensors are composed of two main components: an inactive soft substrate and a deformable conductor. The soft substrates are usually made of elastomers which enable the hyperelasticity for soft sensors. Conductors are used to convey and obtaining information from the environment. There are several kinds of conductors embedded in the soft sensors [26], such as thin metal films [27], liquid metals [28], ionic liquids [29], conductive polymer composites [30], and conductive inks [31]. Some shortcomings of these methods are the complicated manufacturing process and the susceptibility to electromagnetic interference. The soft optical sensors work by detecting the changing in transmission through optical fibers and provide solutions to overcome these shortcomings. Examples of optical sensors include fiber Bragg grating (FBG) sensors [32], stretchable reflective waveguide sensors [33], and soft diffractive optical sensors [34].

Soft sensors can be used to sense curvature [35], pressure [28], strain [31], and their composite [33]. Due to the stretchable feature, soft sensors have promising applications in many fields (shown in Fig. 1.5). A soft wearable motion sensing suit is designed for the lower limb biomechanics measurements [36] (Fig. 1.5. 1). The sensing suit includes three hyperelastic strain sensors which are made of silicone elastomer with channels of liquid metal. A soft artificial skin for hand motion detection is described in Fig. 1.5. 2 [37] for detecting various hand gestures. The joint motions of five fingers are measured by an array of soft strain sensors. Fig. 1.5. 3 shows a biocompatible pressure sensing skins for minimally invasive surgical instru-

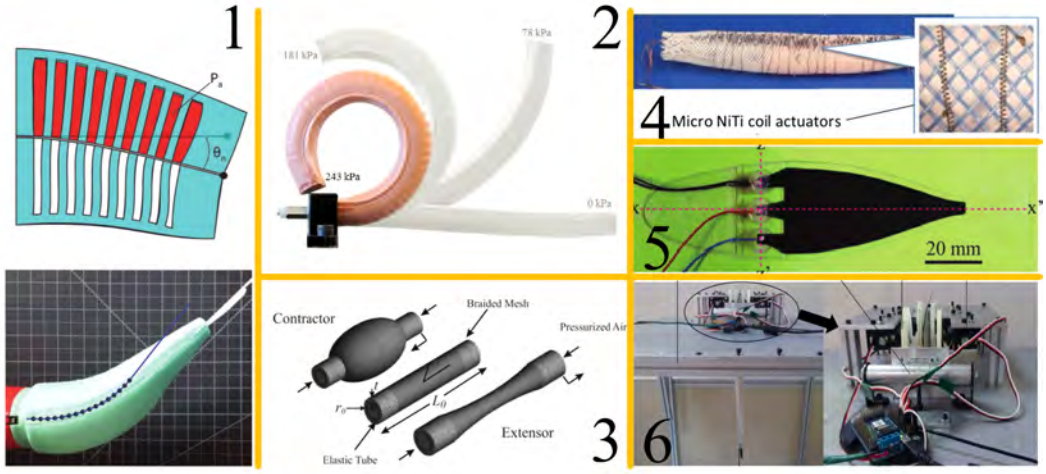

Figure 1.3: Soft actuation technologies: 1. fluidic elastomer actuator [16]; 2. fiber reinforced soft bending actuator [17]; 3. pneumatic artificial muscles [18]; 4. shape memory alloy actuators [19]; 5. dielectric elastomer actuators [20]; 6. cable actuators [21].

ments [38]. The sensor skin is composed of arrays of pressure sensors where the conductive medium uses a biocompatible salt-saturated glycerol solution. A stretchable tactile sensor which is composed of two parallel channel is developed to detect contact force at movable portions is developed in [39] (shown in Fig. 1.5. 4). The contact force can be measured by comparing the resistance changes of each channel. A method for the fabrication of a low-cost, six-axis force-torque sensor (Fig. 1.5. 5) is proposed in [40] using the rubber-encased MEMS barometers. In [41], a soft skin module (Fig. 1.5. 6) with a built-in airtight cavity is developed for the safe human-robot interaction. The air pressure in the sensor cavity can be sensed for contact sensing and gentle grasping. As shown in Fig. 1.5. 7, a fiber optically sensorized robotic hand is designed in [32]. The soft skin embedded by an FBG array is able to estimate the location of the contacts and the FBG sensors on the bones provides force readings of the contacts.

1.2 Motivation and Challenges

Compared with their rigid-bodies counterparts, soft robots have infinite degrees of freedom so that traditional methods to model and control rigid robots are difficult to be implemented for soft robots and thus new approaches are required. Although many design methodologies have been proposed, the accuracy and efficiency of soft robots is still limited by the difficulties of modeling, control and sensing of deformable systems. With such a problem in mind, the

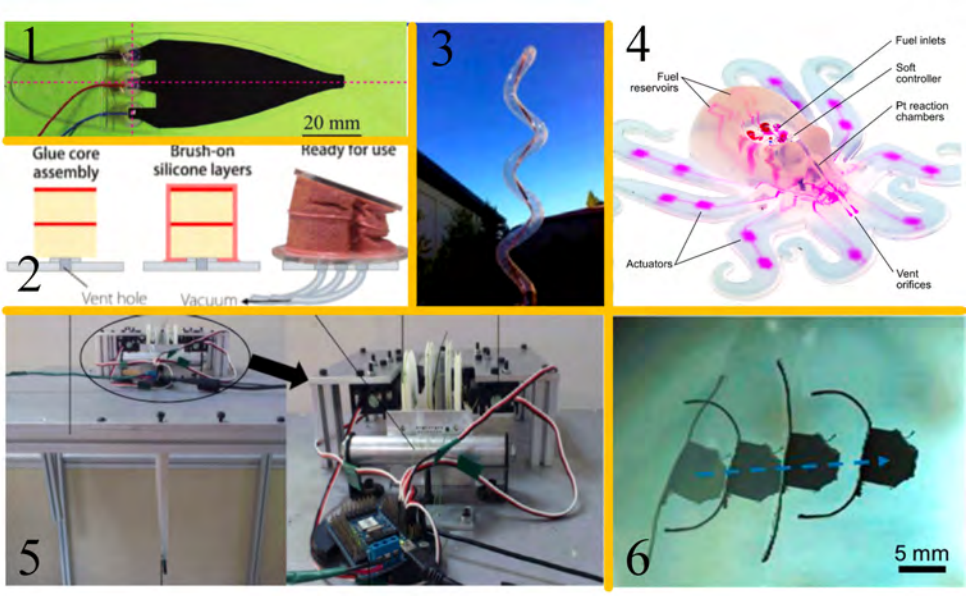

Figure 1.4: Power sources: 1.voltage [20]; 2. vacuum [22]; 3. air pressure [23]; 4. chemical fuel [24]; 5. motor [21]; 6. magnetic field [25].

purpose of this thesis is to explore kinematic modeling and the technologies of vision-based motion control, calibration and force sensing for soft robots.

1.2.1 Modeling

It is usually very difficult to get a relatively accurate model in analytical form for soft robots and it is impossible with complex shapes or contacts with the environment. As a feasible numerical method, Finite Element Method (FEM) is usually limited to be used to provide a more realistic description of mechanics performance due to its higher computational cost. However some work, in particular in medical simulation [42] or computer graphics [43] have proposed real-time implementation of FEM. Thus, real-time FEM provides a general strategy to model the soft robots with a higher accuracy and an acceptable computation time. The quasi-static equation provides the relationship between the actuators and effectors directly.

For the controller design and force sensing, the basic quasi-static equation needs to be extended. Therefore, the thesis introduces two extensions: (1) the discrete-time kinematic equation and Jacobian matrix for kinematic control design, and (2) the equation to describe the relationship between the actuators, external forces and effectors for force sensing.

Traditionally, catheter robots are modeled using the analytical method with an assumption of constant curvature. This method is simple but it is difficult to be employed to model the contacts acting on any position of catheter robot. The loss of contact information might result

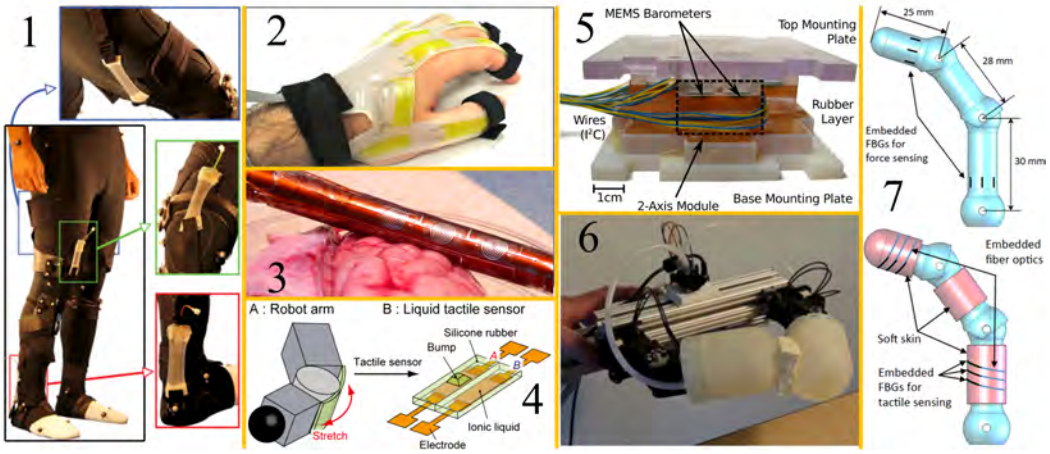

Figure 1.5: Applications of soft sensors. 1: soft wearable motion sensing suit [36]; 2: a soft artificial skin for hand motion detection [37]; 3: biocompatible pressure sensing skins for minimally invasive surgical instruments [38]; 4: a stretchable tactile sensor for the detection of contact force at movable portions [39]; 5: a six-axis force-torque sensor [40]; 6: 3D printed soft skin for safe human-robot interaction [41]; 7: a fiber optically sensorized robotic hand [32].

in imprecise steering and even the failure of the task because of the unsensed Jacobian matrix rotation [44]. Therefore, there is a need to explore new method to model catheter robots interacting with the environment.

1.2.2 Calibration

Robotic calibration can be a key requirement to increase the accuracy of position control and force sensing. Indeed, the robot model can be improved with calibrated sensor-robot system, geometric and material parameters. These items are coupled between each other and can be simultaneously calibrated. The calibration of rigid have been widely researched. However, there are only few work about the calibration of soft robots.

Compared to rigid robots, the calibration of soft robots is more complex due to the difficulty to model the deformation mechanics of a soft material. If soft robots have simple configurations, it is possible to deduce their analytical equations so that the calibration of geometrical and material parameters can be calibrated using the methods for rigid robots. For soft robots with complex configurations, traditional methods are difficult to be employed.

The actuators can also be calibrated using embedded force sensors or position encoders in a decoupled manner with respect to other parameters. However, the use of sensors is expensive and sometimes limited by the space. Therefore, the simultaneous calibration of sensor-robot system and actuators, without using sensors embedded in the actuators, is investigated in this

thesis. By optimizing the difference between feature points on the real robot and its FE model, the position of actuators and the rigid sensor-robot transformation can be calibrated. The strategy is general and can be employed for robots with complex configurations.

1.2.3 Motion Control

Control design based on FE model provides a general strategy to control soft robots. The former work in the team [45–47] is focused on the open-loop control design based on the quasi-static equation of real-time FE model. The control inputs are computed without using feedback position from soft robots, so open-loop controllers have relatively lower accuracy and no robustness to disturbances when the robots are modeled imprecisely. To overcome this drawback, this thesis contributes to develop the closed-loop control of soft robots based on real-time FE model, in particular using vision-based strategy.

Jacobian matrix is estimated using the FE model and is employed for the controller design. If the FE model error is limited in a small range, the control performances can reach to a high level of accuracy with the assumption that the Jacobian matrix is estimated accurately. However, for real applications, a larger estimation error of Jacobian matrix could affect the stability of the closed-loop system so that the robust stability is investigated in this thesis.

Catheter robots have bendable axes which increase the maneuverability at the tip. In the past decade, both manual [48] and automatic steerable systems [49] have been designed to increase the safety and the procedural effectiveness. However, the precise steering using a more accurate, safer, and more reliable approach still remains to be one of the main challenges for catheter insertion. The loss of contact information might result in unsensed Jacobian matrix rotation for a cable-driven continuum catheter robot [44]. Besides, the available force feedback during catheter insertion is an important strategy to avoid damaging the blood vessel by applying excessive force. However, to our best knowledge, none of the available strategies are capable of providing the catheter–tissue interaction forces along the entire catheter. There is, therefore, a need to report sufficient information of contact forces to the surgeon to increase the safety of catheter insertion. The thesis contributes to deal with the two challenging problems and introduce a strategy for the motion control of catheter robot through contacts.

1.2.4 Force Sensing

The property of force sensing increases the autonomy and safe level for object manipulation and medical applications. A general strategy to make the robot feel its environment is achieved by integrating external force sensors into the soft body. However, the integration of external force sensors could increase the complexity and the volume of mechanical structures. To deal

with this drawback, force sensing by the soft robot itself provides a promising strategy for applications with possible image monitoring (for instance: MRI, scan or ultra-sound for medical applications) and limited workspace. Prior works concentrated on the external force computation for continuum robot with contacts at the tip [50, 51]. There is no general strategy which can be employed for soft robots with all the features like complex structure, 3D workspace, force sensing on the whole body, and multi-force actuation.

The idea of this thesis is to show two general vision-based methods (marker-based and marker-free) to solve a much more challenging problem: external force sensing in terms of location and intensity.

For catheter robots, the available force feedback during catheter insertion is an important strategy to avoid damaging the blood vessel by applying excessive force. The contact forces that the catheter tip applies on the blood vessel can be measured by a sensor placed on the catheter tip or by an indirect force measurements (like an image based force sensing). However, none of the available strategies is capable of providing the catheter-tissue interaction forces along the entire catheter. There is, therefore, a need to report sufficient information of contact forces to the surgeon by visual or haptic feedback to increase the safety of catheter insertion.

1.3 Thesis Organization

The thesis is organized into 6 chapters as follows:

Chapter 2 introduces the related work, theoretical background and summarizes the contributions of the thesis.

Chapter 3 first deduces the discrete-time kinematic equation for soft robot based on real-time FE method. Then, Chapter 3 introduces a general method of simultaneous calibration for soft robots. Using the proposed method, both the sensor-robot transmission and the initial input of actuators can be calibrated at the same time. Two closed-loop position control strategies are proposed with the estimation of Jacobian matrix from a simulator. The robust stability is also analysized for the case of imprecise estimation of Jacobian matrix. Finally, by combining both open-loop control and closed-loop control, a switching control strategy is achieved to deal with the problem of image feature loss.

Chapter 4 presents a vision-based force sensing strategy for soft robots with complex configurations. Not only the intensity but also the location can be estimated using several markers on the surface of the robot. In addition, the relationship between the number of markers and the external forces is explored. Finally, evaluation index is proposed to evaluate the distribution of markers and candidate locations, and estimate the maximum error of computed forces.

Chapter 4 also presents a marker-free force sensing approach for soft robot using an RGB-D camera. The locations of external forces are detected based on the segmentation of the point cloud, while the intensities are computed by solving an inverse problem.

Chapter 5 proposes a decoupled control strategy for the motion control of a cable-driven continuum catheter robot through contacts. The proposed method can deal with the problem of Jacobian matrix rotation and allows to compute the contact forces along the entire catheter robot. Besides, a simplified approach is proposed to solving a quadratic programming (QP) problem with a linear complementary constraint (LCC).

Chapter 6 concludes the thesis and shows the future work.

Chapter 2

Related Work and Contributions

In this chapter, the related work about modeling, position control, forces sensing and calibration for soft robots is reviewed. Besides, we introduce the theoretical backgrounds: the quasi-static modeling for soft robots based on the Finite Element Method, contact modeling and the Lyapunov stability theory. The backgrounds are employed for the kinematic modeling, calibration, position control and force sensing in this thesis. Finally, the contributions of this thesis are listed.

2.1 Related Work

In this section, we give an overview about the related work on the modeling, calibration, position control and force sensing for soft robots.

2.1.1 Modeling

Soft robots can be modeled based on either analytical models or numerical models (shown in Fig. 2.1). The popular analytical models are constant curvature model (CCM), non-constant curvature model (NCCM) and Pseudo-Rigid-Body model (PRBM). Numerical models using Finite Element Method (FEM) provide a general framework to model soft robots and other structures with complex configurations. The advantages and disadvantages are summarized in Tab. 2.1.

2.1.1.1 Analytical Models

In the last ten years, researchers have done many works [21, 52] based on the CCM model which can be considered as the simplest model of soft robots. It consists of two separate

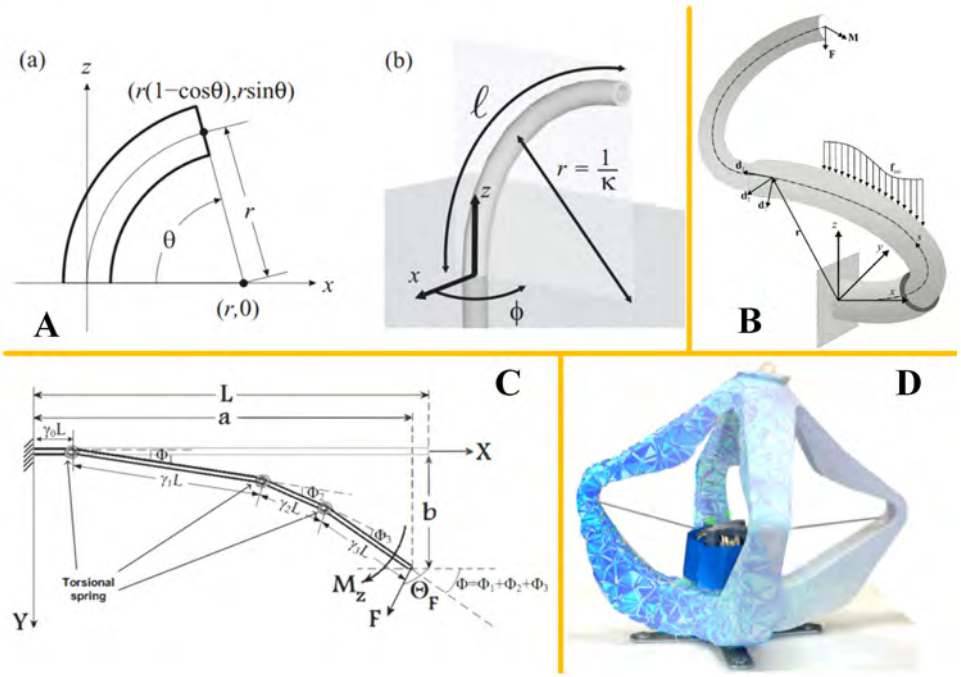

Figure 2.1: Models of soft robots: A. piece-wise constant curvature model [52]; B. non-constant curvature model [53]; C. Pseudo-Rigid-Body model [54]; D. Finite Element model [45].

sub-mappings: one is general to map configuration space to task space, and the other is robot-specific to map actuator space to configuration space. However, in many practical cases, such as the significant contacts with the environment and the non-constant deformation, the CCM does not capture all features of soft robots.

Non-constant curvature models (NCCM) are therefore proposed, with a higher degree of accuracy, to model soft robots. As a kind of non-constant curvature models, Cosserat geometrically exact models based on Cosserat rod theory has been explored for kinematic and dynamic modeling [53, 56, 57, 59]. The NCCM is considered to have a higher accuracy. The deformation is computed by solving a group of differential equations with boundary conditions. The derivation of equations for NCCM needs heavy mathematical skills and the computation is also time-consuming. Besides, control design is difficult because of the complex governing equations and boundary conditions.

Pseudo-Rigid-Body model [54] is also a simple strategy to model continuum robot by converting the flexible link to several rigid links connected by elastic joints.

All these methods assimilate the robot as a curve and suppose that the cross-section is constant.

	model	advantages	disadvantages
analytical	CCM [21, 52, 55]	simple; control design	no external load; only simple deformation
	NCCM [53, 56–60]	external loads	difficult to model; difficult for control design
	PRBM [54, 61]	simple; control design	only 2D; lower accuracy
numerical	FEM [45, 62–68]	general; external loads;	time-consuming

Table 2.1: Summary of existing modeling techniques for continuum and soft robots

2.1.1.2 Numerical Models

The Finite Element Method (FEM) provides an approach to model soft robots with generic assumption on the shape and on the constitutive material of the robots. FEM is often used to provide a more realistic description of mechanics performance [66, 67] due to its higher computational cost. However, it has been proved that we are capable today of computing FE models of soft tissues in real-time for haptic rendering [68]. The modeling of soft robots based on real-time FEM was first proposed in [45] which opens a new research field for real time modeling and control of soft robots.

FEM has been employed to model the contacts between the soft object and the soft or rigid environment [62, 63, 65]. FEM enables real-time computations of contact, cutting and other topological changes for the interactive simulation of surgical procedures.

2.1.1.3 Modeling of Catheters

As a specific application, catheter robots have been modeled by several methods in the literature. One approach employs the pseudo-rigid-body (PRB) model to convert the flexible link to several rigid links connected by elastic joints [54, 61]. In order to capture more accurate deformation, Cosserat rod theory is used to model the 3-D static deflection of catheters [58, 60]. The catheter robot can also be regarded as a continuum robot with the assumption of constant curvature [55]. In [69], the 3-D deflection of a catheter robot is analyzed by discretizing the robot backbone curves to infinitesimal segments. Catheter can also be modeled using beam elements [64]. FEM is an effective strategy to simulate the 3D deformation of structures with external forces. Using the real-time FEM technology, computational cost is acceptable for the interactive simulation.

2.1.2 Calibration

Calibration of rigid robots has been intensively investigated and most of the strategies focus on calibrating the robot geometrical parameters [70] and the robot-sensor system [71]. However, there are only a few works for the calibration of deformable robots. The calibration of soft robots or soft objects is based on either analytical model or numerical model. Several existing works are listed in Tab. 2.2.

model used for calibration	advantages
analytical model [72–75]	more parameters can be calibrated
numerical model [73, 76–80]	objects with complex configurations
model-free [81]	without model

Table 2.2: Summary of existing calibration techniques

Based on a constant curvature model, in [74], the geometrical parameters are calibrated for a continuum robot. A simultaneous calibration of robot-sensor system, geometrical and material parameters for a soft gripper is proposed in [75]. The geometrical and material parameters are calibrated for a concentric tube robot [72]. The stiffness matrix is calibrated for the stiffness control of a continuum robot [73].

If the robot or the object have complex configurations, it is usually difficult to get their analytical model. However, it is possible to model them using Finite Element Method. By minimizing the deflection between loading and without loading, the material parameters of a soft object are calibrated in [73, 76–79].

Model-free calibration is investigated in [81] where a machine learning technique is employed to calibrate a soft motion sensor system.

2.1.3 Position Control

Control design for soft robot can be achieved using model-based or model-free strategies which are summarized in Tab. 2.3.

model based control	CCM [21, 82, 83]
	NCCM [12, 84]
	FEM [45–47, 85, 86]
model free control	PID [26, 66, 87, 88]
	model-less [44, 89, 90]
	learn inverse kinematic equation [91, 92]
	learn the control strategy [93, 94]

Table 2.3: Summary of existing position control techniques

2.1.3.1 Model-based Control

Most model-based control are proposed based on the CCM. In [21], the visual servoing approach is applied for position control of cable-driven soft robotic manipulator based on a CCM model. In [82], a cascaded curvature controller is designed to control a highly compliant 2D manipulator actuated by bi-directional fluidic elastomer actuators. Then the method is extended to control a soft spatial fluidic elastomer manipulator [83].

There is only few work about the control design based on the NCCM [12, 84]. Controller design based on NCCM is more challenging because the boundary conditions should be considered for the control design.

The first application of FEM in control design is shown in [45] where a novel method of modeling and control of soft robots is presented by extracting the reduced mechanical compliance in the space of the actuators and the end-effector. Further improvements and applications of this idea are presented in [46] where an asynchronous simulation framework is proposed to improve the control performances. Without using the feedback information from the robotic system, the above controllers are implemented in open-loop form which results in low control performances when the robots are modeled imprecisely. This observation motivated the work in this thesis using the feedback from the vision systems.

For the vision-based control design, image feature loss is a situation that we should consider. Some existing methods that deal with image feature loss can be found. One strategy is to improve the algorithms to avoid any visual feature loss [95]. Another approach that addresses the problem of image feature loss is based on a prediction of the location of the features [96]. Generally speaking, the methods based on prediction are available to continue the visual servoing task in case of feature loss. However, none of them can effectively deal with image feature loss during a long period of time.

2.1.3.2 Model-free Control

To avoid the difficulties of modeling, model-free control methods were proposed to control soft robots using learning or optimization approaches. Model-less method is proposed in [44, 89] where the Jacobian matrix is updated by optimization using the measured data. There are a number of work where machine learning is used to control soft robots. Some work tried to learn the inverse kinematic equations [91, 92], some tried to learn the control strategies directly [93, 94] using the reinforcement learning technique. Besides, for the control of soft actuators or the manipulator with a single actuator, PID type of control strategy is simple and effective [26, 66, 88]. Generally speaking, model-free controllers have higher robustness to the change of environment but they are relatively complex with lower precision or longer time

for learning.

2.1.3.3 Control of Catheter Robots

Catheter robots can be controlled using the methods for continuum robots. Most position controllers are designed based on the inverse kinematic model so that the computation of inverse Jacobian matrix is indispensable. Jacobian matrix can be computed based on either the analytical model [49, 97] or the numerical model [85, 86]. These methods are proposed for the position control of catheters or soft robots without considering the interaction with the environment. Without an accurate catheter model, the catheter tip can also be controlled by the model-less [90] (Jacobian matrix is estimated empirically using the input and output data) or the model-free [87] control strategies. Although they are robust to the environment, contact forces along the entire robot are difficult to be captured. The contact is modeled as a linear complementarity problem (LCP) [64]. Treating the contact forces as optimization variables is employed for both rigid robot [98] and soft manipulator [47]. However, combining inverse model based on optimization and contact modeling is difficult, in particular in real-time. To our best knowledge, it has only been done previously by a work of our team [47].

2.1.4 Force Sensing

Current researches on force sensing for soft objects can be divided into three major categories (see Tab. 2.4): direct sensing, indirect sensing and computation based on FE method.

methods	applications	advantages
using embedded sensors	grippers [11, 99–101]	independent; accurate
	manipulators and catheters [102–105]	
vision-based force sensing	continuum robots[50, 51, 106–108]	simple; cheap
	catheter robots [109–112]	
	using RGB-D camera [79, 80, 113, 114]	
FEM simulation	catheters [62, 64, 65, 115]	without using sensors

Table 2.4: Summary of existing force sensing techniques

2.1.4.1 Force Sensing using Embedded Sensors

The direct sensing strategy employs embedded sensors to measure the strain directly. Force sensing using embedded force sensors are widely used used not only for rigid robots [116, 117] but also for soft grippers and manipulators.

Resistive bend sensors are embedded in the soft robotic gripper to provide internal state measurements for haptic identification of objects [11]. The stretchable optical waveguides are used for the strain sensing in a prosthetic hand which can feel the shape and softness of an object [99]. In [100], sensor skins are integrated on a soft robotic gripper which enable haptic object visualization. A soft robotic gripper is build using soft somatosensitive actuator with proprioceptive and haptic feedback, which are enabled by embedded curvature, inflation, and contact sensors [101].

In order to measure the contact forces between the catheter tip and the organs, a force sensor is embedded in the catheters (such as Fig. 2.2 A). In [102, 103], an elastic element is embedded in the tip of catheters and optical fibers are used to transmit the reflective light. The contact force can be measured by detecting the optical fiber light intensity. The helically-wrapped fiber Bragg grating (FBG) sensors are embedded in a continuum robot to provide simultaneous curvature, torsion, and force measurement [104]. In [105], the FBG sensors are placed along the arc length of a continuum manipulator for both force and shape sensing.

The direct sensing approaches generally lead to more accurate estimation of external forces. However, mounting sensors onto or inside the manipulator is not only expensive but always limited by the size of the instrument and the softness of the considered robot (as sensors have their own stiffness) [103].

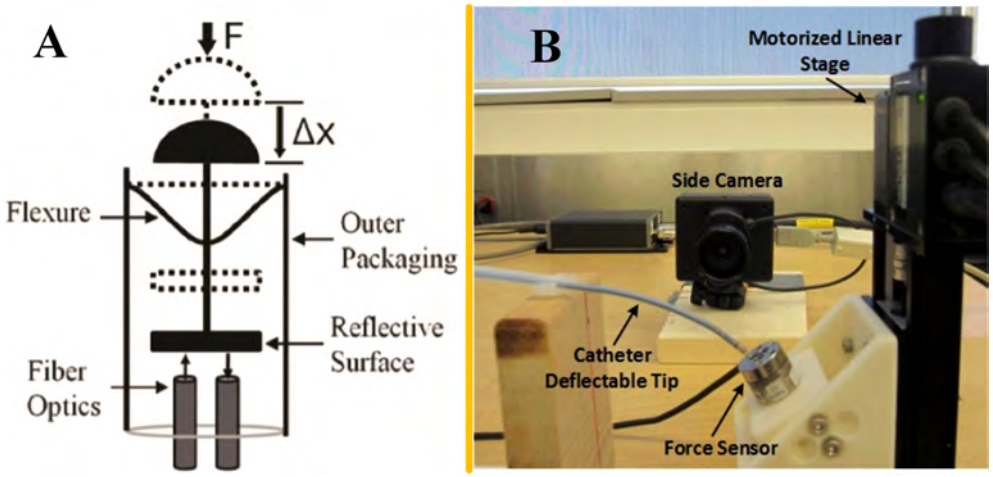

Figure 2.2: Force sensing methods: A. embedded force sensor [103]; B. vision-based force sensing [110].

2.1.4.2 Vision-based Force Sensing

The idea of indirect sensing is to estimate external forces using the soft robot or the soft object itself as a force sensors. The contact forces can be estimated based on the deformation detected

by a camera (as shown in Fig. 2.2 B).

In the field of rigid robotics research, the torque of joint motors can be used to compute the external forces applied on the tip of the robot [118, 119]. For soft robots, the actuator forces can also be employed to estimate the end-effector forces for a continuum manipulator [50] and a parallel continuum robot [51]. There are works about deflection-based force sensing to estimate tip forces. A probabilistic approach is described in [106] to estimate forces based on pose measurements. In [108], a force sensing method is proposed based on the shape reconstruction algorithm and the kinematic–static model. Sensing multiple external forces along the entire material or robot is more challenging because of the estimation of both force intensities and locations. In [107], kinematic-based contact detection and localization methods for multi-segment continuum robots are presented.

There are many works on force sensing for catheter robots using vision-based strategies. Using the information of tip pose and shape measurements, a 3D force estimation platform is designed in [111] for tendon-driven catheters based on the Cosserat rod model and Kalman filter. The contact forces can be estimated based on shape detection [109], tip pose measurement [112] and shaft curvature [110]. All these approaches for catheter robots are limited to sense load just on the catheter tip.

RGB-D cameras have been used to detect the location of external forces for rigid robots [113] and to estimate contact forces for hand-object manipulation [114]. In [79], the external force acting on a soft object is computed through the registration of an FE model on point cloud data provided by an RGB-D sensor. The force sensing approaches based on the analytical models are difficult to be used for soft robots with complex geometrical shapes. However, FEM provides possibilities to sense external forces for soft objects which have complex configurations.

2.1.4.3 Force Sensing based on FEM Simulation

Given the known environment and catheter model, the location of contact forces is detected by collision detection algorithm and the intensity along the entire catheter can be computed by the Finite Element Method [64]. This strategy is widely used for medical simulation where the contact is modeled based on the Signorini's law [62, 65]. Using this force sensing approach, we need to have accurate models of catheter and its contacts.

2.2 Theoretical Background

We briefly introduce the theoretical backgrounds on quasi-static modeling, contact modeling, linear complementarity problem and Lyapunov theorem. The backgrounds in this section are

employed for the work in the following chapters.

2.2.1 Quasi-static Equations for Soft Robots based on FEM

This subsection is built on the modeling method in [45] which propose to build the equilibrium function for soft objects in the constraint space. The models proposed by this paper describe the evolution of the steady state (position of each nodes of the FEM) of the system when a given input variation is applied. The are expressed in the form of implicit discrete-time equations.

At each sampling time k, the quasi-static equilibrium function (i.e for low velocities) of the entire robot is given by [1]:

$$\mathbf{f}_{ext,k+1} - \mathbf{f}(\mathbf{x}_{k+1}) + \mathbf{H}_{a,k+1}^T \lambda_{a,k+1} + \mathbf{H}_{c,k+1}^T \lambda_{c,k+1} = 0 \tag{2.1}$$

where $\mathbf{x}$ is the position vector of all FEM nodes and is updated according to the configuration of the soft robots at each sampling time. The directions of the forces on the nodes exerted by actuators contacts are proved by $\mathbf{H}_a^T$ and $\mathbf{H}_c^T$. λ_a and λ_c are the force of actuators and contacts respectively. Therefore, $\mathbf{H}_a^T \lambda_a$ and $\mathbf{H}_c^T \lambda_c$ represent the contributions of the actuators and the contacts respectively to the position of FEM nodes. $\mathbf{f}_{ext}$ represents the external loads (like the gravity and the contact forces) and $\mathbf{f}(\mathbf{x})$ gathers the internal stiffness forces.

A linearization of the internal forces is computed at the $(k+1)th$ sampling time using the FE model:

$$\mathbf{f}(\mathbf{x}_{k+1}) \approx \mathbf{f}(\mathbf{x}_k) + \mathbf{K}(\mathbf{x}_k)\, d\mathbf{x}_{k+1} \tag{2.2}$$

where $\mathbf{K}(\mathbf{x}_k)$ is the tangent stiffness matrix that depends on the actual positions of the nodes and $d\mathbf{x}_{k+1}$ is the displacement between consecutive positions ($d\mathbf{x}_{k+1} = \mathbf{x}_{k+1} - \mathbf{x}_k$). Substituting (2.2) into (2.1), the equilibrium equation at each step can be established:

$$\mathbf{K}(\mathbf{x}_k)\, d\mathbf{x}_{k+1} = \mathbf{f}_{ext,k+1} - \mathbf{f}(\mathbf{x}_k) + \mathbf{H}_{k+1}^T \lambda_{k+1} \tag{2.3}$$

In motion space, the matrix $\mathbf{K}$ is often very large in dimension so that the direct computation of equilibrium equation is expensive. Instead, using the Schur complement of the constraint problem, the FE model equation can be projected into the constraint space that

[1] This function is built on the assumption of low velocities, so the dynamic feature (inertia force) of the robot is ignored.

drastically reduces its size. Then, we have:

$$\delta_{e,k} = \mathbf{W}_{ea}(\mathbf{x}_{k-1})\lambda_{a,k} + \mathbf{W}_{ec}(\mathbf{x}_{k-1})\lambda_{c,k} + \delta_{e,k}^{free} \tag{2.4}$$

$$\delta_{a,k} = \mathbf{W}_{aa}(\mathbf{x}_{k-1})\lambda_{a,k} + \mathbf{W}_{ac}(\mathbf{x}_{k-1})\lambda_{c,k} + \delta_{a,k}^{free} \tag{2.5}$$

$$\delta_{c,k} = \mathbf{W}_{ca}(\mathbf{x}_{k-1})\lambda_{a,k} + \mathbf{W}_{cc}(\mathbf{x}_{k-1})\lambda_{c,k} + \delta_{c,k}^{free} \tag{2.6}$$

where $\mathbf{W}_{ij}(\mathbf{x}_k) = \mathbf{H}_i(\mathbf{x}_k)\mathbf{K}^{-1}(\mathbf{x}_k)\mathbf{H}_j^T(\mathbf{x}_k)$ with $i, j = e, a; e, c; a, a; a, c; c, a$ and c, c. δ_e and δ_a are, respectively, positions of the end effector and the actuator in constraint space. δ_e^{free} and δ_a^{free} are positions given at the free configuration ($\lambda_a = 0$).

$\mathbf{W}_{ij}$ are the basic matrices and are homogeneous to a compliance. Using the compliance matrices, we can get a measure of the mechanical coupling between effectors, actuators and contacts. The matrix $\mathbf{K}$ can be factorized using a *LDL* decomposition, even for a large number of nodes and actuators.

At the $(k+1)$ *th* step, (2.4) can be written as:

$$\mathbf{x}_{k+1} = \mathbf{K}_k^{-1}\mathbf{H}_a^T \cdot \lambda_k + \mathbf{x}_k^{free} \tag{2.7}$$

2.2.2 Contact Modeling

To model the contact between the soft robots with rigid or soft environment, one needs first to detect the collision. Collision detection is used to detect the intersection of two or more objects [120]. It has been a fundamental problem in physical simulations, computer animation and robotics.

The collision response is based on the Signorini's law [121]. For every contact point, there is a complementarity relation between the interpenetration gap δ_c and the contact force λ_c among the normal direction. The complementarity relation can be written as [62]:

$$\mathbf{0} \leq \delta_c \perp \lambda_c \geq \mathbf{0} \tag{2.8}$$

where $\perp$ is the complementarity operator. It states that one of the two values δ_c or λ_c mush be null.

2.2.3 Linear Complementarity Problem

The Linear Complementarity Problem (LCP) is defined as follows [122]:

Definition (The Linear complementarity problem): Given a matrix $M \in \mathbb{R}^{n\times n}$ and a vector $q \in \mathbb{R}^n$, find two vectors $\omega, z \in \mathbb{R}^n$ such that

$$\omega = Mz + q \tag{2.9}$$

$$\omega,\ z \geq 0 \tag{2.10}$$

$$z^T \omega = 0 \tag{2.11}$$

The complementarity condition (2.11) is equivalent to $z_i\omega_i = 0$ for $i = 1,\cdots,n$. For all i, at most one of ω_i and z_i can be positive.

The complementarity constraints (2.10) and (2.11) can also be expressed as $0 \leq z \perp \omega \geq 0$. The LCP can also be formulated as:

$$Mz + q \geq 0 \tag{2.12}$$

$$z \geq 0 \tag{2.13}$$

$$z^T (Mz + q) = 0 \tag{2.14}$$

The contact equations (2.6) and (2.8) can be formulated as a linear complementarity problem:

$$\mathbf{W}_{ca}\lambda_a + \mathbf{W}_{cc}\lambda_c + \delta_c^{free} \geq 0 \tag{2.15}$$

$$\lambda_c \geq 0 \tag{2.16}$$

$$\lambda_c^T \left(\mathbf{W}_{ca}\lambda_a + \mathbf{W}_{cc}\lambda_c + \delta_c^{free}\right) = 0 \tag{2.17}$$

2.2.4 Lyapunov Theorem

Lyapunov theorems establish very general sufficient conditions to analyze the stability of dynamic systems. Before the introduction of Lyapunov theorems, we first introduce the basic concepts of equilibrium point, stability and asymptotic stability for the following autonomous

system:

$$\dot{\mathbf{x}} = f(\mathbf{x}) \tag{2.18}$$

Definition (Equilibrium Point): A state $\mathbf{x}^*$ is an equilibrium state (or equilibrium point) of the system (2.18) if $f(\mathbf{x}^*) = 0$

Definition (Stability): The equilibrium state $\mathbf{x}^* = 0$ of the system (2.18) is stable if for any $R > 0$, there exists $r > 0$, such that

$$\| \mathbf{x}(0) \| < r \implies \| \mathbf{x}(t) \| < R, \quad \forall t \geq 0$$

Otherwise, the equilibrium point is unstable.

Definition (Asymptotic Stability): An equilibrium point $\mathbf{0}$ is asymptotically stable if it is stable and there exists some $r > 0$, such that

$$\| \mathbf{x}(0) \| < r \implies \lim_{t \to \infty} \mathbf{x}(t) = 0$$

The positive definite functions are defined as:

Definition (Positive Definite): A scalar continuous function $V(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$ is said to be positive definite if $V(\mathbf{0}) = 0$ and $V(\mathbf{x}) > 0, \ \forall \mathbf{x} \neq 0$; $V(\mathbf{x})$ is said to be positive semidefinite if $V(\mathbf{0}) = 0$ and $V(\mathbf{x}) \geq 0, \ \forall \mathbf{x} \neq 0$; $V(\mathbf{x})$ is said to be negative definite (negative semidefinite) if $-V(\mathbf{x})$ is positive definite (positive semidefinite).

Lyapunov theorems about the local stability and global stability for continuous systems and discrete-time systems are listed as follows [123]:

Theorem (Local Stability): Assume that there exists a scalar function V of the state $\mathbf{x}$, with continuous first order derivatives such that

- $V(\mathbf{x})$ is positive definite
- $\dot{V}(\mathbf{x})$ is negative definite

then the equilibrium at the origin is asymptotically stable.

Theorem (Global Stability): Assume that there exists a scalar function V of the state $\mathbf{x}$, with continuous first order derivatives such that

- $V(\mathbf{x})$ is positive definite
- $\dot{V}(\mathbf{x})$ is negative definite
- $V(\mathbf{x}) \to \infty$ as $\| \mathbf{x} \| \to \infty$

then the equilibrium at the origin is globally asymptotically stable.

Essentially the results hold for the discrete-time system

$$x(k+1) = f(x(k)) \tag{2.19}$$

If $V(\mathbf{x})$ is positive definite and $\triangle V(\mathbf{x}) = V(f(\mathbf{x})) - V(\mathbf{x})$ is negative definite, the origin is asymptotically stable.

2.3 Contribution of the Thesis

This thesis presents innovative work on vision-based calibration, closed-loop position control and external force sensing for soft robots. Five main contributions are addressed as following:

1. Based on real-time FEM, the discrete-time kinematic equation of soft robot is deduced to map between actuator space and task space. The Jacobian matrix of soft robots is estimated using simulation-based predictor and is employed for the closed-loop control design. Considering the model error, the robust stability of the closed-loop system is analyzed based on Lyapunov stability theory. Combining the open-loop controller and the closed-loop controller, a switched control method is proposed to deal with the problem of image features loss.

2. Based on the inverse quasi-static equation of soft robots, the simultaneous calibration of sensor-robot system and initial actuator inputs is converted as an optimization problem. The objective is to minimize the distance of feature points on soft robot and its FE model. Then, we employ two methods (particle swarm optimization and sequential optimization) to compute the parameters.

3. The thesis proposes a marker-based external force sensing method for soft robots using a vision system. The implementation is based on the displacement of feature points defined on soft robot and its FE model. The thesis shows the proof that both the intensity and the locations of external forces can be estimated only using the feature points on soft robot. Then, a strategy is proposed to find the correct locations of external forces among several possible ones.

4. The marker-free vision-based force sensing strategy is proposed using point cloud generated by an RGB-D camera. This strategy can realize both the location detection and intensity computation of multiple external forces without using a force sensor.

5. The thesis presents a safer strategy of catheter insertion by independent control of translation and bending as well as visualization of contracts forces along the catheter. A closed-loop control strategy is proposed for the bending control of catheter robot with contacts being taken into consideration. The control inputs are computed by solving a QP problem with

LCP. A simplified strategy to solve this problem is proposed and the LCP is proved to hold strictly for reachable trajectory tracking.

2.4 List of Publications

Papers under review

- **Z. Zhang**, A. Petit, J. Dequidt, and C. Duriez, "Calibration and External Force Sensing for Soft Robots using an RGB-D Camera," IEEE Robotics and Automation Letters, (Under review).

Journal papers

- **Z. Zhang**, J. Dequidt, J. Back, H. Liu and C. Duriez, "Motion Control of Cable-Driven Continuum Catheter Robot through Contacts," IEEE Robotics and Automation Letters (RA-Letter), 2019 (accepted).

- **Z. Zhang**, J. Dequidt, and C. Duriez, "Vision-Based Sensing of External Forces Acting on Soft Robots Using Finite Element Method," IEEE Robotics and Automation Letters (RA-Letter), 2018.

- T. Bieze, A. Kruszewski, F. Largilliere, **Z. Zhang**, R. Merzouki, C. Duriez. "FEM-based kinematics and closed-loop control of soft, continuum manipulators". Soft Robotics, 2018.

- E. Coevoet, T. Morales-Bieze, F. Largilliere, **Z. Zhang**, M. Thieffry, M. Sanz-Lopez, B. Carrez, D. Marchal, O. Goury, and J. Dequidt, "Software toolkit for modeling, simulation, and control of soft robots," Advanced Robotics, vol. 31, no. 22, pp. 1208–1224, 2017. (Best paper award)

Conference papers

- **Z. Zhang**, T. Bieze, J. Dequidt, A. Kruszewski, and C. Duriez, "Visual servoing control of soft robots based on finite element model," in Proc. 2017 IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), 2017, pp. 2895–2901.

- **Z. Zhang**, J. Dequidt, A. Kruszewski, F. Largilliere, and C. Duriez, "Kinematic modeling and observer based control of soft robot using real-time finite element method," in Proc. 2016 IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), 2016, pp. 5509–5514.

- C. Duriez, E. Coevoet, F. Largilliere, T. Morales-Bieze, **Z. Zhang**, M. Sanz-Lopez, B. Carrez, D. Marchal, O. Goury, and J. Dequidt, "Framework for online simulation of soft robots with optimization- based inverse model," in Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR), IEEE International Conference on , pp. 111–118, IEEE, 2016. (Best paper finalist)

Workshop presentation

- **Z. Zhang**, J. Dequidt, C. Duriez. "External Force Computation for Soft Robot based on Finite Element Model". Soft Morphological Design for Haptic Sensation, Interaction and Display , in IROS 2017.

Chapter 3

Kinematic Modeling, Calibration and Closed-Loop Position Control

3.1 Overview

In this chapter, a general discrete-time kinematic model for soft robots with a wide variety of configurations is derived based on the real-time Finite Element (FE) method. Then, a simultaneous calibration of the soft robot is proposed so that both the sensor-robot system and the initial position of the actuators can be calibrated. Combining the soft robotic system and its simulation model, two closed-loop controllers are designed and the condition of robust stability is provided when the estimated Jacobian matrix is not accurate. By combining the closed-loop control and open-loop control, a switched control is also proposed to deal with the image feature loss.

The methods in this chapter are based on our papers [85], [86] and [80]. To the best of our knowledge, [85] is the first paper to deduce the kinematic equation, Jacobian matrix and closed-loop position control for soft robots based on the real-time Finite Element method. [86] is the first paper to analyze the robust stability of soft robots which are controlled based on the real-time FE model and to design a switched controller for soft robots.

This chapter is structured as follows. Section 2 introduces a discrete-time kinematic model and the features of Jacobian matrix for soft robots based on real-time Finite Element Method. A simultaneous calibration method is presented in Section 3 for the calibration of both the sensor-robot system and the initial actuator inputs. The simulation-based predictor of Jacobian matrix and open-loop control framework is introduced in Section 4. Two closed-loop position control frameworks are shown in Section 5 and Section 6. Section 7 presents a switched control strategy for soft robots. The experimental results on a parallel soft robot are shown in

Section 8. Finally, the conclusion and discussion are presented in Section 9.

3.2 Kinematic Modeling based on Finite Element Method

In this section, the kinematic modeling for soft robots are investigated based on the quasi-static equation introduced in Chapter 2. Then, the Jacobian matrix and the features of Jacobian matrix are discussed.

3.2.1 Kinematic Modeling

Without considering the contacts, we rewrite the quasi-static equation introduced in Chapter 2 as follows:

$$\delta_{e,k} = \mathbf{W}_{ea}(\mathbf{x}_{k-1})\,\lambda_{a,k} + \delta_{e,k}^{free} \tag{3.1}$$

$$\delta_{a,k} = \mathbf{W}_{aa}(\mathbf{x}_{k-1})\,\lambda_{a,k} + \delta_{a,k}^{free} \tag{3.2}$$

At the $(k+1)$ *th* step, (3.1) can be written as:

$$\delta_{e,k+1} = \mathbf{W}_{ea}(\mathbf{x}_{k})\,\lambda_{a,k+1} + \delta_{e,k+1}^{free} \tag{3.3}$$

Combining (3.1) and (3.3), and considering the reasonable assumption of small displacements between two consecutive steps, i.e. $\mathbf{W}_{ea}(\mathbf{x}_k) \approx \mathbf{W}_{ea}(\mathbf{x}_{k-1})$ and quasi-static motion $\delta_{e,k}^{free} \approx \delta_{e,k+1}^{free}$, we obtain the discrete-time model of soft robots:

$$\delta_{e,k+1} = \delta_{e,k} + \mathbf{W}_{ea}(\mathbf{x}_k)\,\triangle\lambda_{a,k} \tag{3.4}$$

where $\triangle\lambda_{a,k} = \lambda_{a,k+1} - \lambda_{a,k}$ is the incremental force contribution of actuators.

Similarly, from (3.2), we have

$$\delta_{a,k+1} = \delta_{a,k} + \mathbf{W}_{aa}(\mathbf{x}_k)\,\triangle\lambda_{a,k} \tag{3.5}$$

3.2.2 Jacobian Matrix

In the research field of robotics, Jacobian Matrix (represents the differential relationship between the actuator displacement and the effector motion) is heavily used to analyze and control

robots. For rigid robots, assuming that the relationship between the task space and joint space is $\mathbf{x} = f(\mathbf{q})$, the kinematics model of the robot can be written as:

$$d\mathbf{x} = \mathbf{J}d\mathbf{q} \tag{3.6}$$

where $\mathbf{J}$ is the Jacobian Matrix and can be expressed as:

$$\mathbf{J} = \frac{df(\mathbf{q})}{d\mathbf{q}} \tag{3.7}$$

For rigid robots, $\mathbf{J}$ is deduced based on the geometric models and depends only on actuator position $\mathbf{q}$. However, for soft robots, $\mathbf{J}$ is deduced based on both the geometrical and material properties and depends on the positions of actuators and all the other degrees of freedom. For the control of soft robotic systems, the position of end effector δ_e is the output variable in task space and δ_a is the input signal to the actuator in joint space. Combining kinematic equations (3.4) and (3.5), the differential relationship for soft robots can be written as:

$$d\delta_e = \mathbf{W}_{ea}\mathbf{W}_{aa}^{-1} \cdot d\delta_a \tag{3.8}$$

where $d\delta_e = \delta_{e,k+1} - \delta_{e,k}$ and $d\delta_a = \delta_{a,k+1} - \delta_{a,k}$. The Jacobian Matrix for soft robots is $\mathbf{J} = \mathbf{W}_{ea}\mathbf{W}_{aa}^{-1}$.

3.2.3 Features of the Jacobian Matrix

The features of $\mathbf{J}(\mathbf{x})$ depend on those of the compliance matrices $\mathbf{W}_{ea}(\mathbf{x})$ and $\mathbf{W}_{aa}(\mathbf{x})$. Two features needed for the controller design are:

(1) Non constant matrices. The compliance matrices $\mathbf{W}_{ea}(\mathbf{x})$ and $\mathbf{W}_{aa}(\mathbf{x})$ cannot be pre-computed because their values depend on the positions of all nodes (the configuration of soft robot) so that they change at each iteration. Based on the fact that the soft materials have slow dynamics and are not subject to high-frequency external loads, we make the assumption that the matrices $\mathbf{W}_{ea}(\mathbf{x})$ and $\mathbf{W}_{aa}(\mathbf{x})$ are constant between two sampling times.

(2) Rank of the matrix. The tangent stiffness matrix $\mathbf{K}(\mathbf{x})$ is positive definite when the robot is constrained to have no rigid body motion. By placing the actuators on different nodes of the FE model or with different directions, there will be no linear dependency between lines of $\mathbf{H}_a$ and $\mathbf{H}_e$ so that the matrices $\mathbf{W}_{aa}(\mathbf{x})$ is positive-definite and $\mathbf{W}_{ea}(\mathbf{x})$ is generally full row rank when the number of lines in $\mathbf{H}_e$ is smaller than the number of lines in $\mathbf{H}_a$, and $\mathbf{W}_{ea}(\mathbf{x})$ is full column rank in the opposite case.

3.3 Calibration

For the closed-loop control and force sensing based on FE model, the calibration between the robot and the FE model is indispensable to make sure that the initial configurations are same and are measured with respect to the same coordinate system. To simplify the calibration method, we assume that both the geometric and material parameters of the soft robot are modeled accurately. The calibration items are thus the rigid transformation between the robot and the FE model, and the initial actuator inputs of FE model.

As an off-line procedure, we consider the use of fiducial markers as feature points placed on the robot surface. Their corresponding positions (δ_{robot}) are defined as effectors for the FE model with respect to the robot frame and are matched with the feature points (δ_{cam}) in the camera frame. The goal of the calibration technique is then to minimize the following position error with respect to the camera/robot rigid transformation $(\mathbf{R},\mathbf{T})$ and the initial actuator input λ_a:

$$\begin{aligned} \min_{\mathbf{R},\mathbf{T},\lambda_a} & \sum_{i=1}^{n} \| \mathbf{R}\delta^i_{cam} + \mathbf{T} - \delta^i_{robot} \|^2 \\ \text{s.t.} \quad & \lambda_a \in \Omega \end{aligned} \tag{3.9}$$

with $\delta_{robot} = \mathbf{W}_{ea}\lambda_a + \delta_e^{free}$. $\| * \|_2$ is the 2-Norm and Ω is the constraint for actuator input. n is the number of feature points. $\mathbf{T} \in \mathbb{R}^3$ and $\mathbf{R} \in \mathbb{R}^{3\times3}$ are respectively the translation vector and the rotation matrix to align each observed marker to their positions in the robot base frame defined for the FE model.

The rigid transformation can be written as

$$\mathbf{T} = \begin{bmatrix} X_0 & Y_0 & Z_0 \end{bmatrix} \tag{3.10}$$

and

$$\begin{aligned} \mathbf{R} &= Rot(Z,\alpha)\,Rot(Y,\beta)\,Rot(X,\gamma) \\ &= \begin{bmatrix} c_\alpha c_\beta & c_\alpha s_\beta s_\gamma - s_\alpha c_\gamma & c_\alpha s_\beta c_\gamma + s_\alpha s_\gamma \\ s_\alpha c_\beta & s_\alpha s_\beta s_\gamma + c_\alpha c_\gamma & s_\alpha s_\beta c_\gamma - c_\alpha s_\gamma \\ -s_\beta & c_\beta s_\gamma & c_\beta c_\gamma \end{bmatrix} \end{aligned} \tag{3.11}$$

with s_* and c_* being the shorthands of $\sin *$ and $\cos *$.

In order to compute the calibrated parameters based on (3.9), we employ two methods:

simultaneous optimization and sequential optimization.

3.3.1 simultaneous optimization

The calibration variables consist of six parameters (X_0, Y_0, Z_0, α, β, γ) for rigid transformation and parameters (the elements in vector λ_k) to control the deformation.

Particle Swarm Optimization (PSO) is employed to solve the optimization problem (3.9). Inspired by a bird flock or a fish school, PSO (see [124] for a tutorial) is a global gradient-free optimization algorithm, which has has been employed for many applications [125].

In PSO, the candidate solutions of the optimization problem are defined as particles. The position $\mathbf{w}_i$ and velocity $\mathbf{s}_i$ of the particles are updated based on their own and neighbors' best positions.

$$\mathbf{s}_i(t+1) = \omega(t)\,\mathbf{s}_i(t) + c_1\,(p_i - \mathbf{w}_i(t))\,R_1 + c_2\,(g_i - \mathbf{w}_i(t))\,R_2 \tag{3.12}$$

$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t) + \mathbf{s}_i(t+1) \tag{3.13}$$

where c_1 and c_2 are two acceleration constants. p_i is the best solution which has achieved so far by particle i, while g_i is the global best solution obtained by the swarm. R_1 and R_2 are two random numbers between 0 and 1 and they the same for each particle. $\omega(t)$ is a the inertia weight and, in this chapter, the linearly decreasing inertia weight [126] is employed:

$$\omega(t) = \omega_{start} - \frac{\omega_{start} - \omega_{end}}{t_{max}} t \tag{3.14}$$

where ω_{start} and ω_{end} are respectively the initial and final values of the inertia weight. t_{max} and t are the maximum number of allowable iterations and the current iteration times, respectively.

In (3.9), $\mathbf{W}_{ea,k}$ and $\delta_{e,k}^{free}$ cannot be computed based on the initial configuration of FE model. Therefore, during the calibration, the configuration of FE model should be updated and then new values of $\mathbf{W}_{ea,k}$ and $\delta_{e,k}^{free}$ can be computed. The optimization problem (3.9) is solved using PSO after each update of FE model.

The calibration is finished if the number of update reaches the predefined number or the calibration error reaches its minimum value. Both the predefined number and the minimum value are robot-specific. For each update of FE model, PSO is running several times to make sure that we can get the optimal solution. The calibration error is regarded to reach the minimum value if it can not be reduced any more after several updates of FE model. If the difference for the initial actuator inputs is small (the configuration of the soft robot is close to its FE model), (3.9) can be solved without the update of FE model.

3.3.2 sequential optimization

The PSO optimization method is difficult to get the global optimal solution so that we have to use a larger number of particles and iterations, which make the calibration time-consuming. We propose a faster strategy to solve the optimization problem (3.9). The method consists of two alternated parts: the computation of the optimal rigid transformation and the optimization of the actuator inputs.

The correspondences between two sets of points (the feature points on the robot surface and the effectors on the FE model) generates a linear least-squares problem that can be solved robustly using the Singular Value Decomposition (SVD) method. The optimal rigid transformation can be obtained by minimizing the following error:

$$\min_{\mathbf{R},\mathbf{T}} \sum_{i=1}^{n} \| \mathbf{R}\delta^i_{cam} + \mathbf{T} - \delta^i_{robot} \|^2 \tag{3.15}$$

Where n is the number of markers on the robot surface. Based on the optimal rigid transformation, we can compute the transformed positions of markers using

$$\delta^i_{tran} = \mathbf{R}\delta^i_{cam} + \mathbf{T} \tag{3.16}$$

The actuator input is then computed by minimizing the distance between the feature points on the FE model and the transformed markers:

$$\begin{aligned} &\min_{\lambda_a} \frac{1}{2} \| \mathbf{W}_{ea}\lambda_a + \delta^{free}_e - \delta_{tran} \|^2 \\ &\text{s.t.} \quad \lambda_a \in \Omega \end{aligned} \tag{3.17}$$

It can be converted to a standard quadratic programming (QP) formulation

$$\begin{aligned} &\min_{\lambda_a} \frac{1}{2}\lambda_a^T \mathbf{Q}\lambda_a + \mathbf{c}^T\lambda_a \\ &\text{s.t.} \quad \lambda_a \in \Omega \end{aligned} \tag{3.18}$$

where $\mathbf{Q} = \mathbf{W}^T_{ea}\mathbf{W}_{ea}$ and $\mathbf{c} = \mathbf{W}^T_{ea}\left(\delta^{free}_e - \delta_{tran}\right)$.

The calibration algorithm is also shown in (3.3.2). In (3.9), $\mathbf{W}_{ea}$ and δ^{free}_e cannot be precomputed based on the initial configuration of the FE model. Therefore, after the computation of λ_a, the configuration of the FE model should be updated with I_2 iterations in order to

compute more accurate values for $\mathbf{W}_{ea}$ and δ_e^{free}. The calibration is finished if the iteration reaches the defined value I_1.

Algorithm 3.1 Calibration algorithm using sequential optimization

1. Define corresponding feature points
2. Read positions of feature points from sensors
3. While $Iteration < I_1$
    Compute positions of feature points (3.4)
    Compute rigid transformation **T** and **R** (3.15)
    Compute positions of transformed feature points (3.16)
    While $Iteration < I_2$
        Compute actuator forces λ_a (3.18)
        Update the configuration of FE model
4. Output T, **R** and λ_a

3.4 Simulation-based Predictor and Open-loop Control

For rigid robots, that have generally few Degrees of Freedom (DoF), Jacobian matrix $\mathbf{J}(\mathbf{x})$ can be computed easily using the measurement of several state variables $\mathbf{x}$ from the real robots. However, the way we obtain $\mathbf{J}(\mathbf{x})$ for soft robots, as presented above, is different. For soft robots, the large number of states $\mathbf{x}$ can not be fully measured from the robot directly.

Thanks to the development of the real-time FEM technology, soft robots can be simulated in real-time to get the value of $\mathbf{J}(\mathbf{x})$ at each sampling time after the configuration has been updated. A closed-loop controller for the simulation model can be designed based on (3.4) to make sure that the Jacobian matrix of the simulation model stays close to that of the real robot. Besides, the simulation-based predictor can also generate an open-loop controller for soft robots.

As shown in Fig. 3.1, the implementation of open-loop controller consists of two parts: closed-loop control of simulation model and the open-loop control of soft robot. We still define the strategy as open-loop control because no feedback information from the soft robot is needed. In this strategy, both the simulation model and the soft robot share the same control input which is the contribution of the actuators and is computed based on the simulation model.

Using the implementation of open-loop control, the soft robot can be controlled without using external sensors. However, the performances of open-loop control deteriorate if the FE model is not accurate. In order to get a high control accuracy, it is necessary to close the control loop using external sensors.

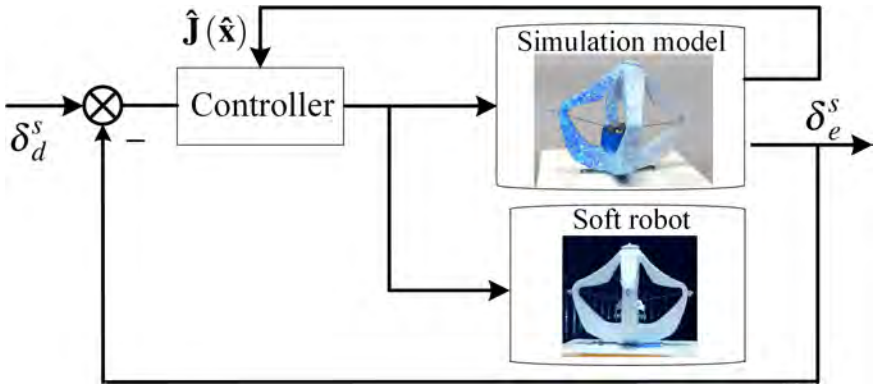

Figure 3.1: The implementation of the open-loop controller. At each sampling time, $\hat{\mathbf{J}}(\hat{\mathbf{x}})$ is computed from the simulation model and is used for the computation of control input for both the simulation model and soft robot.

3.5 First Implementation of Closed-Loop Control

3.5.1 Introduction

The objective of this section is to design a control law so that the end effector of the robot δ_e can reach its desired position $\delta_{e,d}$ asymptotically. Unlike the methods to compute control inputs using Gauss-Seidel [45] or Quadratic Programming [127] algorithms based on quasi-static equation (3.1), the controller is designed based on the kinematic equation (3.4) and is implemented using a closed-loop strategy.

In traditional control implementation, control inputs are computed by control law which is only based the information in the real space. This is a feasible way to design controllers whose feedback information can be measured or observed in application (usually it is for systems with simple models). In this section, we introduce a novel implementation of predictor based control for complex systems. The principle of this control strategy is shown in Fig. 3.2.

The FE model serves as a predictor to obtain the values of matrices $\mathbf{W}_{ea}(\mathbf{x}_k)$ and $\mathbf{W}_{aa}(\mathbf{x}_k)$ at each sampling time. The change of actuation $\triangle\lambda_a$ is computed using information from two basic components: one is the predictor built by the Finite Element Model (to obtain matrix $\mathbf{W}_{ea}$) and the other is the robotic system (to obtain the output δ_e). The input signal δ_a which is computed by substituting $\triangle\lambda_a$ and $\mathbf{W}_{aa}$ into (3.5) is used to actuate the real robot.

3.5.2 Control based on Discrete-time Model

Three assumptions are made to simplify the design process.

1. The FE model of soft robots is a precise model of the real robot and constitutes an accurate state predictor;

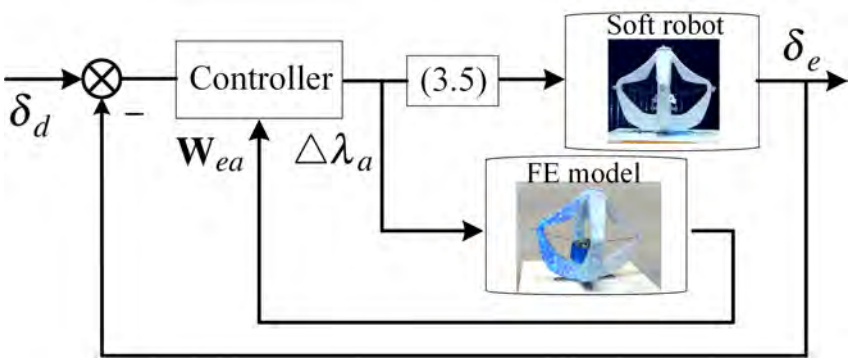

Figure 3.2: The first implementation of closed-loop control. Information from both the soft robot and the FE model is needed to compute the control inputs. The FE model provides the value of compliance matrix in (3.4) for the controller.

2. The robot workspace contains no singular configuration;
3. The actuators are not constrained (no saturation).

Based on the Lyapunov method, the position control law $\triangle\boldsymbol{\lambda}_a$ for soft robot is designed and is shown in the following theorem:

Theorem 1: Considering the closed-loop system, described by (3.4), the discrete-time feedback control law is designed as

$$\triangle\boldsymbol{\lambda}_{a,k} = -\mu\mathbf{W}_{ea}^{+}(\mathbf{x}_k)\left(\delta_{e,k} - \delta_{e,d}\right) \tag{3.19}$$

where $\delta_{e,k}$ is the position of effector measured from the real robot and $\delta_{e,d}$ is the desired position for the effector of real robot. Then, for any μ s.t. $0 < \mu < 2$, the system is globally asymptotically stable.

Proof: The error signal $\mathbf{e}$ is defined as: $\mathbf{e} = \delta_e - \delta_{e,d}$. We consider the Lyapunov candidate function:

$$V = \mathbf{e}^T\mathbf{e} \tag{3.20}$$

The variation of Lyapunov function can be written as:

$$\begin{aligned}
\triangle V &\triangleq V(k+1) - V(k) \\
&= \mathbf{e}_{k+1}^T\mathbf{e}_{k+1} - \mathbf{e}_k^T\mathbf{e}_k \\
&= \left(\delta_{e,k+1} - \delta_{e,d}\right)^T\left(\delta_{e,k+1} - \delta_{e,d}\right) - \left(\delta_{e,k} - \delta_{e,d}\right)^T\left(\delta_{e,k} - \delta_{e,d}\right) \\
&= \delta_{e,k+1}^T\delta_{e,k+1} - \delta_{e,k}^T\delta_{e,k} - 2\delta_{e,k+1}^T\delta_{e,d} + 2\delta_{e,k}^T\delta_{e,d}
\end{aligned} \tag{3.21}$$

Substituting (3.4) into (3.21), we have

$$\triangle V = \left(2\delta_{e,k}^T + \left(\mathbf{W}_{ea}\left(\mathbf{x}_k\right)\triangle\lambda_{a,k}\right)^T - 2\delta_{e,d}^T\right)\mathbf{W}_{ea}\left(\mathbf{x}_k\right)\triangle\lambda_{a,k} \tag{3.22}$$

It is assumed that the $\mathbf{W}_{ea}\left(\mathbf{x}_k\right)$ of the real robot is the same as the that of the FE model, so we have $\mathbf{W}_{ea}\left(\mathbf{x}_k\right)\mathbf{W}_{ea}^{+}\left(\mathbf{x}_k\right) = \mathbf{I}$.

Then, substituting (3.19) into (3.22), we have

$$\triangle V = -\mu\left(2\delta_{e,k}^T - 2\delta_{e,d}^T - \mu\left(\delta_{e,k} - \delta_{e,d}\right)^T\right)\left(\delta_{e,k} - \delta_{e,d}\right) \tag{3.23}$$

Finally, $\triangle V$ can be written as

$$\triangle V = -\mu\left(2-\mu\right)\mathbf{e}_k^T\mathbf{e}_k \leq 0 \tag{3.24}$$

In (3.24), $\triangle V < 0$ holds for all $\mathbf{e} \neq 0$, so that all solutions converge to the set $S := \left\{\delta_e \mid \delta_e\left(k\right) = \delta_{e,d}\right\}$. From (3.19), we can further conclude that the input $\triangle\lambda_{a,k}$ converges to zero, so that (3.5) is also stable.

If the number of actuators is equal to the dimension of end effector space, we have $\mathbf{W}_{ea,k}^{+} = \mathbf{W}_{ea,k}^{-1}$ and get the unique solution of control inputs. Otherwise, the least 2-norm solution of the control input can be obtained using the control law (3.19).

3.6 Second Implementation of Closed-Loop Control

In the previous section, the FE model is still controlled using open-loop strategy. The FE model could diverge from the real robot if there are interactions with the environment. In this section, we will also update the FE model to avoid the potential divergence.

3.6.1 Introduction

The second implementation of closed-loop control is illustrated in Fig. 3.3.

Controller 1 is designed to make sure that the end effector of soft robot tracks its desired trajectory δ_d. Controller 2 is employed to actuate the simulation model so that the end effector of simulation model follows the end effector of the real robot. Thanks to controller 2, the configurations of both simulation model and soft robot are similar. We assume that there is no significant perturbation on the robot (strong contact or force) which would create dissimilar configurations between the real and simulated robot, while having a good registration of the effector. Therefore, the Jacobian matrix of soft robot can be predicted by the simulation model.

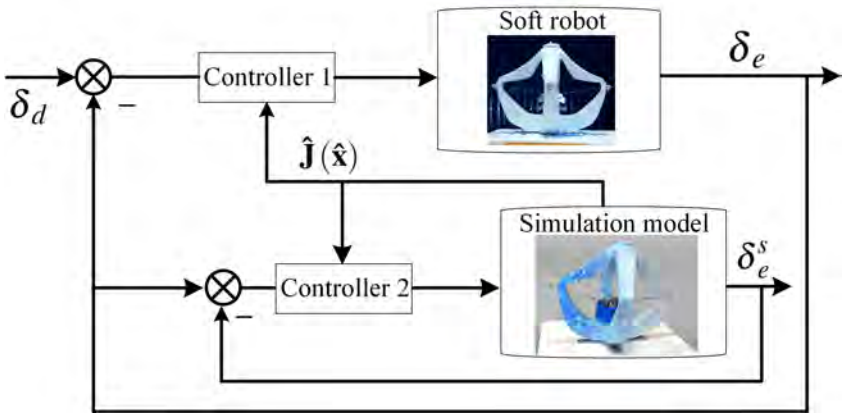

Figure 3.3: The second implementation of closed-loop control. Controller 1 and Controller 2 are the closed-loop controllers for the soft robot and the simulation model, respectively. The two controllers can be designed independently.

3.6.2 Closed-Loop Control Design

The control design is based on proportional control strategy and pseudo-inverse control allocation. In order to simplify the controller design for trajectory tracking, two assumptions are required:

1. $rank\hat{\mathbf{J}}(\hat{\mathbf{x}}) = 3$ which means that there is no singular configuration in the workspace.
2. The actuators are not constrained (no saturation).

By defining a new control vector $U_k = \hat{\mathbf{J}}(\hat{\mathbf{x}}) \triangle\delta_{a,k}$, the kinematic model can be rewritten as

$$\delta_{e,k+1} = \delta_{e,k} + U_k \tag{3.25}$$

The tracking error is defined as $e_k = \delta_{e,k} - \delta_{d,k}$. In the task space, the control vector U_k can be designed as:

$$U_k = -k_p e_k - \left(\delta_{d,k} - \delta_{d,k+1}\right) \tag{3.26}$$

where k_p is a constant parameter for the proportional gain.

Then the Pseudo-inverse based control allocation is employed to obtain a unique solution:

$$u = \hat{\mathbf{J}}^{+}(\hat{\mathbf{x}}_k) U_k \tag{3.27}$$

which is the minimum 2-norm solution of $U_k = \hat{\mathbf{J}}(\hat{\mathbf{x}}_k) \triangle\delta_{a,k}$ and $u = \triangle\delta_{a,k}$ is the control input of the robot. For over-actuated system, the contribution of actuators with minimal energy can be computed by (3.27). Considering that the matrix $\hat{\mathbf{J}}(\hat{\mathbf{x}})$ has full row rank, the Pseudo-inverse

matrix $\hat{\mathbf{J}}^{+}(\hat{\mathbf{x}})$ can be computed by $\hat{\mathbf{J}}^{+}(\hat{\mathbf{x}}) = \hat{\mathbf{J}}^{T}(\hat{\mathbf{x}}) \left(\hat{\mathbf{J}}(\hat{\mathbf{x}}) \hat{\mathbf{J}}^{T}(\hat{\mathbf{x}})\right)^{-1}$.

Substituting (3.26) into the (3.27), the control law (for controller 1) can be written as:

$$u = -\hat{\mathbf{J}}^{+}(\hat{\mathbf{x}}_k) \left(k_p e_k + \delta_{d,k} - \delta_{d,k+1}\right) \tag{3.28}$$

We use the same implementation strategy for the control law of controller 2, concerning the simulated robot:

$$u^s = -\hat{\mathbf{J}}^{+}(\hat{\mathbf{x}}_k) \left(k_p^s e_k^s + \delta_{d,k}^s - \delta_{d,k+1}^s\right) \tag{3.29}$$

where k_p^s is also a constant parameter for the proportional gain.

3.6.3 Closed-Loop Stability Analysis

In this subsection, we consider the situation where the Jacobian matrix is estimated imprecisely, i.e. $\hat{\mathbf{J}}(\hat{\mathbf{x}}) \neq \mathbf{J}(\mathbf{x})$. The object of the analysis is to get the range of control parameter k_p to make sure that the closed-loop system is stable. Considering that the closed-loop system is linear, the stability analysis is done with $\delta_{d,k} = \delta_{d,k+1}$.

Substituting (3.28) into (3.8), we have the closed-loop system:

$$e_{k+1} = (1 - k_p) e_k + \omega_k \tag{3.30}$$

where

$$\omega_k = k_p \left[\mathbf{I} - \mathbf{J}(\mathbf{x}_k) \hat{\mathbf{J}}^{+}(\hat{\mathbf{x}}_k)\right] e_k$$

We assume that, even if the Jacobian matrix is estimated imprecisely, the error is bounded. By introducing a scalar γ, we have

$$\omega_k^T \omega_k \leq \gamma^2 k_p^2 e_k^T e_k \tag{3.31}$$

where

$$\gamma^2 \geq \| \left[\mathbf{I} - \mathbf{J}(\mathbf{x}_k) \hat{\mathbf{J}}^{+}(\hat{\mathbf{x}}_k)\right]^T \left[\mathbf{I} - \mathbf{J}(\mathbf{x}_k) \hat{\mathbf{J}}^{+}(\hat{\mathbf{x}}_k)\right] \| \tag{3.32}$$

and $\gamma \geq 0$.

In order to keep the robust stability of the closed-loop system (3.30), we have the following theorem:

Theorem 2: The robust stability of the closed-loop system (3.30) stands if

$$k_p \in \left(0, \quad \frac{2}{1+\gamma} \right)$$

where $0 \leq \gamma < 1$.

Proof: We consider the Lyapunov canditate function:

$$V(e) = e^T e \tag{3.33}$$

In the following, we will find the conditions so that the variation of Lyapunov function is negative:

$$\begin{aligned} \mathbf{d}V \overset{\triangle}{=} & V(e_{k+1}) - V(e_k) \\ = & \mathbf{e}_{k+1}^T \mathbf{e}_{k+1} - \mathbf{e}_k^T \mathbf{e}_k \\ = & \left[(1-k_p)^2 - 1 \right] e_k^T e_k + 2(1-k_p) e_k^T \omega_k + \omega_k^T \omega_k \end{aligned} \tag{3.34}$$

In (3.34), we add the term $-\alpha \left[\omega_k^T \omega_k - \gamma^2 k_p^2 e_k^T e_k \right] + \alpha \left[\omega_k^T \omega_k - \gamma^2 k_p^2 e_k^T e_k \right]$ where $\alpha \geq 0$ is an accessorial parameter for the proof. Then we have

$$\begin{aligned} \mathbf{d}V = & \left[\left(1+\alpha\gamma^2\right) k_p^2 - 2k_p \right] e_k^T e_k + 2(1-k_p) e_k^T \omega_k \\ & + (1-\alpha) \omega_k^T \omega_k + \alpha \left[\omega_k^T \omega_k - \gamma^2 k_p^2 e_k^T e_k \right] \end{aligned} \tag{3.35}$$

Given that $\omega_k^T \omega_k \leq \gamma^2 k_p^2 e_k^T e_k$, we can find a bound for $\mathbf{d}V$. (3.35) can be rewritten into a matrix form:

$$\mathbf{d}V \leq \begin{bmatrix} e_k \\ \omega_k \end{bmatrix}^T \mathbf{B} \begin{bmatrix} e_k \\ \omega_k \end{bmatrix} \tag{3.36}$$

where

$$\mathbf{B} = \begin{bmatrix} \left(1+\alpha\gamma^2\right) k_p^2 - 2k_p & 1-k_p \\ 1-k_p & 1-\alpha \end{bmatrix}$$

To make sure that $\mathbf{d}V < 0$, the matrix $\mathbf{B}$ should be negative definite so that

$$\left(1+\alpha\gamma^2\right) k_p^2 - 2k_p < 0 \tag{3.37}$$

and

$$\begin{aligned}\det(\mathbf{B}) &= \left[\left(1+\alpha\gamma^2\right)k_p^2 - 2k_p\right](1-\alpha) - (1-k_p)^2 \\ &= \left(\alpha\gamma^2 - \alpha - \alpha^2\gamma^2\right)k_p^2 + 2\alpha k_p - 1 > 0\end{aligned} \tag{3.38}$$

From (3.37) and (3.38), we have $1-\alpha < 0$.

The analytical solution of (3.37) and (3.38) is obtained from the following analysis.

From (3.37), we have $0 < k_p < \frac{2}{1+\alpha\gamma^2}$. In (3.38), $\alpha\gamma^2 - \alpha - \alpha^2\gamma^2 < 0$ because $\alpha > 1$ and $\gamma > 0$. In order to make sure that (3.38) has solution, we need the discriminant to be positive, i.e. $\alpha\left(1-\gamma^2\right)(\alpha-1) > 0$ so that $\gamma < 1$.

(3.38) can be rewritten in α as

$$-\gamma^2 k_p^2\alpha^2 + k_p\left(k_p\gamma^2 - k_p + 2\right)\alpha - 1 > 0 \tag{3.39}$$

The dominant term in (3.39) is negative. We need to have real roots, which means that the discriminant must be positive:

$$k_p^2\left(\gamma^2-1\right)\left(\gamma^2k_p^2 - k_p^2 + 4k_p - 4\right) > 0 \tag{3.40}$$

With the condition of $k_p > 0$ and $\gamma < 1$, (3.40) can be simplified as

$$\left(\gamma^2-1\right)k_p^2 + 4k_p - 4 < 0 \tag{3.41}$$

so that we have $k_p < \frac{2}{1+\gamma}$ or $k_p > -\frac{2}{\gamma-1}$. Considering that $\alpha > 1$ and $0 < \gamma < 1$, we have $\frac{2}{1+\alpha\gamma^2} < 2$ and $-\frac{2}{\gamma-1} > 2$ so that there is no intersection between $k_p < \frac{2}{1+\alpha\gamma^2}$ and $k_p > -\frac{2}{\gamma-1}$. Therefore, the solution of (3.41) is $k_p < \frac{2}{1+\gamma}$.

Now we check if there is intersection between the solution set of (3.39) and $\alpha > 1$. The solution of (3.39) is $\alpha_2 < \alpha < \alpha_1$ where α_1 and α_2 are the roots of the polynomial in (3.39). α_1 is computed as

$$\frac{\gamma^2k_p - k_p + 2 + \sqrt{\gamma^4k_p^2 - 2\gamma^2k_p^2 + 4\gamma^2k_p + k_p^2 - 4\gamma^2 - 4k_p + 4}}{2\gamma^2k_p}$$

The control parameter k_p should make both (3.38) ((3.39)) and $\alpha > 1$ hold. To make sure that (3.39) holds with $\alpha > 1$, we need $\alpha_1 > 1$, i.e.

$$f(\gamma, k_p) < \sqrt{\gamma^4k_p^2 - 2\gamma^2k_p^2 + 4\gamma^2k_p + k_p^2 - 4\gamma^2 - 4k_p + 4} \tag{3.42}$$

where $f(\gamma, k_p) = \gamma^2k_p + k_p - 2$. The solution of $f(\gamma, k_p) > 0$ is $k_p > \frac{2}{\gamma^2+1}$. Because $0 <$

$\gamma < 1$, $\frac{2}{\gamma^2+1} > \frac{2}{1+\gamma}$. Therefore, there is no intersection between the solution set of (3.41) and $f(\gamma, k_p) > 0$. On the other hand, $f(\gamma, k_p) \leq 0$ implies $k_p \leq \frac{2}{\gamma^2+1}$ which is compatible with the constraint $k_p < \frac{2}{1+\gamma}$, given that $0 \leq \gamma < 1$. Therefore, if we chose k_p such that $0 < k_p < \frac{2}{1+\gamma}$, $\mathbf{B}$ is negative definite and $\mathbf{d}V < 0$. In that case, the control will be stable.

3.7 Switched Control for Image Feature Loss

A common failure using visual servoing strategies is caused by the image feature loss when the feedback information is not available in applications. To deal with this problem, we introduce a switching control strategy. The switched controller consists of an open-loop controller, a closed-loop controller and switching conditions. If the image feature is available, the closed-loop controller shown in Fig. 3.3 is employed. When the image feature loss occurs, the controller switches to the open-loop strategy shown in Fig. 3.1.

At the last sampling time before the switching time, the contributions of the actuators computed by controller 1 and controller 2 (both are shown in Fig. 3.3) are δ_a and δ_a^s respectively. When the image loss occurs, the incremental contribution of the actuators is $\triangle\delta_a^s$ so that the control inputs for both systems are $\delta_a^s + \triangle\delta_a^s$. It is noted that the control inputs $(\delta_a^s + \triangle\delta_a^s)$ and δ_a for the soft robot at the switching time are different. To avoid abrupt motions at the switching time, the input $\delta_{a,k+1}$ of the soft robot progressively switches from one input to the other:

$$\delta_{a,k+1} = \delta_{a,k} + p\left(\delta_{a,k}^s + \triangle\delta_{a,k}^s - \delta_{a,k}\right) \tag{3.43}$$

where p is a constant parameter ($0 < p < 1$). This parameter can be used to tune the length of the switching time period.

Remark: Intuitively, if the closed-loop subsystem is activated long enough with slow switching (i.e., long enough dwell or average dwell time), the energy increase caused by switching or unstable subsystem can be trade off to maintain the stability of the system [128]. For the implementation of the controller, we assume that the image loss happens occasionally so that switching frequency is low.

3.8 Experiment Results

In this section, we first show the calibration performances. Then, the experimental setup for the closed-loop control is introduced. Finally, we present the control performances of two closed-loop control and the switched control.

3.8.1 Calibration Performances

The calibration technique is validated by both simulation and real experiments. We employ two methods (simultaneous optimization and sequential optimization) to solve the optimization problem (3.9). The calibrated variables in this paper are $\mathbf{R}$, $\mathbf{T}$ and λ_a in (3.9).

As shown in Fig. 3.4, two cases with different number and different distribution of markers are considered for the validation. In the first case, five markers (the white dots in Fig. 3.4) are defined. Nine feature points (both black and white dots in Fig. 3.4) are employed in the second case. The objective of the calibration technique is to actuate the initial configuration (Fig. 3.4 (a)) with respect to *oxyz* in order to reach the desired one (Fig. 3.4 (b)) with respect to *pxyz*. The calibrated configuration is shown in Fig. 3.4 (c) with respect to *oxyz*.

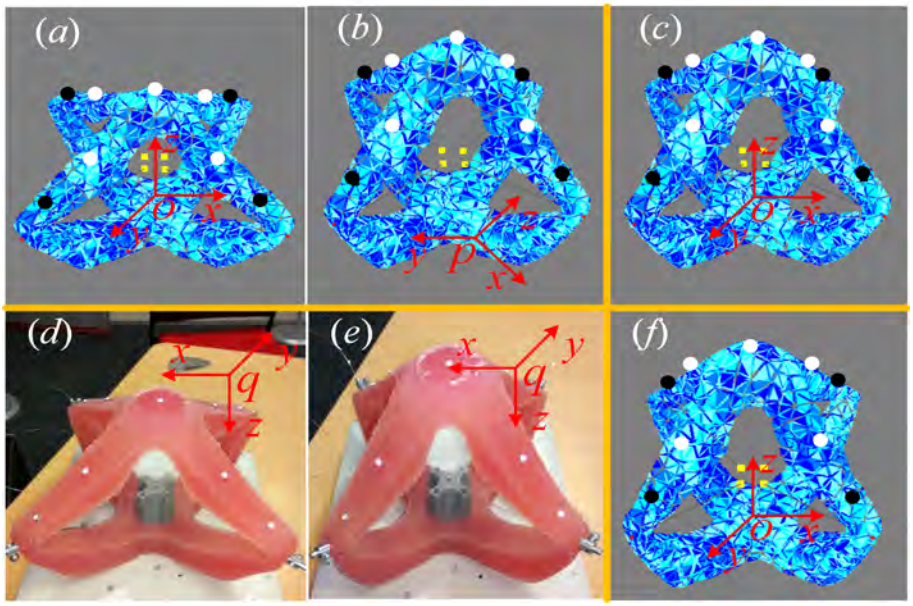

Figure 3.4: Validation design for the calibration by simulation ((a), (b) and (c)) and real experiment ((d), (e) and (f)). (a), (b) and (c) are respectively the initial, the desired and the calibrated (based on (b)) configurations of FE model with respect to *oxyz*, *pxyz* and *oxyz*. (d) and (e) are respectively the initial and the desired configurations of soft robot with respective to *qxyz*. (f) is the calibrated (based on (e)) configuration of the FE model with respect to *oxyz*. The spots (white and black) are the effectors on the FE model.

3.8.1.1 Validation by Simulation

For the implementation of the calibration, it is needed to choose several feature points on the surface of soft robot and define the corresponding points on the surface of FE model.

As shown in the Tab. 3.1, two sets of optimization variables are chosen arbitrarily for the validation. Using the chosen variables, 3D position of feature points are computed based on the direct FE model. Then, the variables are calibrated based on the inverse FE model to make sure that the feature points can reach the same position computed by the direct FE model.

The actual values of all variables for both cases are shown in Table. 3.1 (simultaneous optimization) and Table. 3.3 (sequential optimization). The calibrated values and the residual errors of the objective function (3.9) are listed in Table. 3.1 and Table. 3.3. Being validated by simulation, the proposed calibration strategy using both optimization methods can provide calibrated parameters with a high accuracy.

3.8.1.2 Validation by Real Experiment using Simultaneous Optimization

Instead of using the direct FE model, the real soft robot is employed for the validation. Feature points can consist in artificial markers or obvious feature points, like corners and sharp points, on its surface. Here, we show the validation results in two cases: (1) calibration using five artificial markers and (2) calibration using five artificial markers and four obvious feature points (shown in Fig. 3.4). For both cases, the position of feature points are measured at the same configuration of the soft robot so that the calibration parameters have the same theoretical values if there is no measurement error.

Table. 3.2 shows the residual errors for both cases. The residual error can be reduced to 4.66 *mm* for the first case and 12.82 *mm* for the second case. It is noted that the larger error in the second case does not mean the calibration is worse. The residual error is the sum of position errors for all feature points. From the experiments, the calibrated parameters are also close for both cases.

3.8.1.3 Validation by Real Experiment using Sequential Optimization

For the calibration using sequential optimization, the residual errors with respect to the number of iterations is shown in Fig. 3.5. The simulation-based calibration with nine markers has a faster convergence and a higher accuracy so that we use it for the real experiment.

Instead of using the direct FE model, the real soft robot (Fig. 3.4 (d), (e)) is employed for validation. Here, we show the validation results using nine markers which are selected and manually measured from the point cloud. Fig. 3.4 (a) is also the initial configuration of the FE model. Fig. 3.4 (e) and (f) are respectively the configuration of soft robot with respect to *qxyz* and the calibrated configuration of the FE model with respect to *oxyz*.

Table 3.1: Validation of the proposed calibration method by simulation. The unite of α, β, γ and $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ are radian and newton, respectively. The unite of both X_0, Y_0, Z_0 and the calibration error is millimeter.

parameters		α	β	γ	X_0	Y_0	Z_0	λ_1	λ_2	λ_3	λ_4	calibration error
actual value	Fig. **??** (b)	1.00	1.50	2.00	100.00	200.00	300.00	0.90	1.00	1.10	1.20	
	Fig. **??** (c)	1.50	1.30	0.50	150.00	50.00	400.00	0.50	0.80	1.00	0.60	
calibrated value	Fig. **??** (b)	1.02	1.50	2.02	99.98	200.01	300.02	0.89	0.97	1.10	1.25	$0.028(mm)$
	Fig. **??** (c)	1.50	1.30	0.50	150.00	49.98	400.01	0.50	0.80	1.00	0.60	$0.020(mm)$

Table 3.2: Validation of the proposed calibration method by real experiment. The unite of α, β, γ and $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ are radian and newton, respectively. The unite of both X_0, Y_0, Z_0 and the calibration error is millimeter.

parameters	α	β	γ	X_0	Y_0	Z_0	λ_1	λ_2	λ_3	λ_4	calibration error (mm)
Case 1 (Fig. **??** (e))	0.43	-3.14	0.008	0.44	50.89	562.33	1.28	2.26	1.19	1.76	4.66
Case 2 (Fig. **??** (f))	0.44	-3.14	0.017	-8.18	56.10	564.32	1.31	1.90	1.46	1.83	12.82

Table 3.3: Calibration results using simulation. The unite of λ is newton. The unite of both **T** and the residual error is millimeter. RE: residual error; (I): the validation with five markers; (II) the validation with nine markers.

Items	**R**			**T**	λ
Actual Value	−0.147 −0.485 −0.862	0.941 0.200 −0.273	0.305 −0.851 0.427	−264.956 263.885 12.767	1.200 1.000 1.000 1.200
$RE(I)$ 0.025	−0.147 −0.484 −0.862	0.941 0.200 −0.273	0.305 −0.852 0.426	−264.965 263.913 12.947	1.199 1.000 0.999 1.200
$RE(II)$ 0.024	−0.147 −0.485 −0.862	0.941 0.200 −0.273	0.305 −0.851 0.427	−264.938 263.899 12.799	1.200 1.001 1.001 1.201

Table 3.4 shows the residual error for the real experiment. The initial configuration of the soft robot (Fig. 3.4 (d)) is employed to get the actual value of **R**, **T**. The actual value $\delta_a = \delta_a^{initial} + \triangle\delta_a$ where $\delta_a^{initial}$ is computed by the calibration based on the initial configuration of the soft robot (Fig. 3.4 (d)) and $\triangle\delta_a$ is randomly set by users. Actuated with δ_a, the configuration of soft robot is shown in Fig. 3.4 (e).

As shown in Fig. 3.5, the residual error can be reduced to 7.47 *mm* after ten iterations in our experiment.

Table 3.4: Calibration results using real experiment. The unite of **T**, δ_a, and the residual error is millimeter. AV: actual value; RE: residual error; (II) the validation with nine markers.

Items	**R**			**T**	δ_a
AV	−0.732 0.681 0.000	0.679 0.729 −0.087	−0.059 −0.064 −0.996	17.480 12.598 566.918	3.947 5.176 7.920 7.644
$RE(II)$ 7.47	−0.736 0.677 0.018	0.674 0.735 −0.069	−0.060 −0.039 −0.997	17.338 −1.566 570.349	3.249 5.065 10.921 8.904

It is noted that the residual errors in both Table. 3.2 and Table 3.4 are larger than errors in simulation. This is due to errors in the FE model, in the image processing step and in

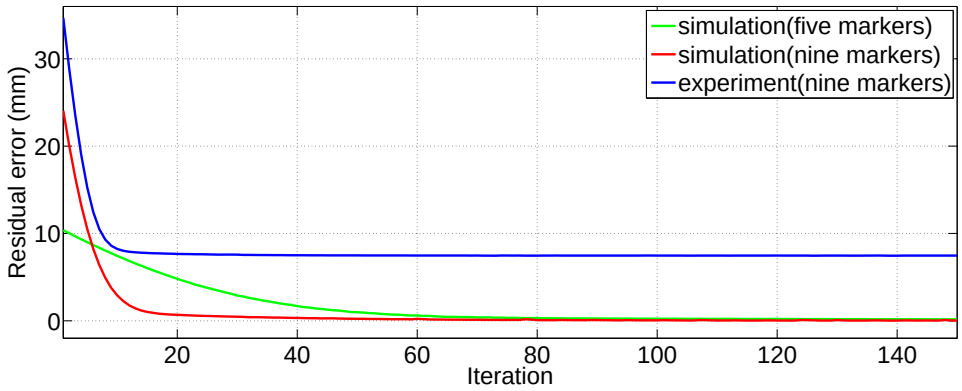

Figure 3.5: Residual errors of the validation with respect to the iteration by simulation and real experiment.

the correspondences between the artificial markers and the feature points defined on the FE model.

The calibration method has several limitations: (1) A limited number of feature points should be defined for the FE model to balance the computation time and external force sensing accuracy; (2) The material parameters needs to be calibrated beforehand; (3) The manual selection of the feature points in the point cloud of the soft robot results in errors to find the corresponding points on the FE model.

3.8.2 Experimental Setup for the Closed-Loop Control

The experimental setup for the visual servoing control of soft robot is depicted in Fig. 3.6. The system consists of a soft robot, a simulation model, and four cameras for the position perception. We use a commercial tracking system *(OptiTrack* by NaturalPoint company with sampling frequency 100Hz and a precision of 0.1mm) to track the 3-D position of end effector.

As a simulation-based predictor, the simulation model is built in SOFA framework and its configuration is updated with the frequency of 31 Hz (Intrl(R) Core(TM) i7-6700HQ CPU @ 2.60Ghz). A volumetric mesh is generated based on the geometry of the robot. The mesh used for this study is composed by 4147 tetrahedra. The optimal number of elements in the mesh cannot be determined analytically, and it is usually chosen as a trade-off between accuracy and computation time. The soft robot is a parallel robot with four soft legs and is actuated by cables. Without using any rigid components, the robot is made of silicone with Young modulus being 150kPa (white robot in the left figure) and 240kPa (pink robot in the right figure). At each sampling time, the displacements of cables are computed by the control law

and are sent to servomotors. In the experiments, the workspace of the robot is limited into a range where the forces of cables can be computed to be positive values.

Figure 3.6: Experimental setup for the position control of soft robots. Two soft parallel robots with different Young modulus (150kPa for the white robot in the left figure and 240kPa for the pink robot in the right figure) are tested. The left robot is used for the experiment of the first closed-loop control and the right one for the second implementation. Both robots have four soft legs and are actuated by cables. The simulation model servers as a simulation-based predictor. The 3D position of the end effector can be obtained by the position perception system.

3.8.3 The First Implementation of Closed-Loop Control

The positioning and tracking performances of open-loop control and two closed-loop control are shown in this section. The position errors of the end effector between its real position and desired position at every step are depicted in Fig. 3.7, Fig. 3.8, and Fig. 3.9, where the red lines depict the changes of position errors using closed-loop controllers while the green lines show that using open-loop controllers.

3.8.3.1 Positioning

In positioning experiments, we consider the performances of both the closed-loop controller and the open-loop controller in two situations: with and without disturbance. The unknown disturbance is generated by an unmodeled rigid ball hung on the robot tip (see Fig. 3.6). The initial tensions of four cables are set as 0.58 N which drives the end effector to the initial position $[3,-122,129]$ (mm) for the robot without disturbance and $[4,-121,111]$ (mm) with disturbance. The desired position is $[10,-110,160]$ (mm) for each case.

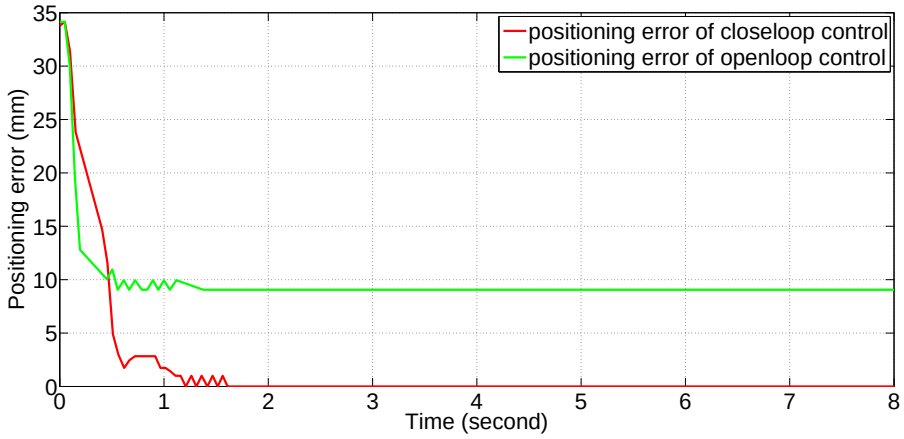

Figure 3.7: Comparison of positioning errors between open-loop control and closed-loop control

The Fig. 3.7 shows the performances of open-loop control and closed-loop control without the disturbance in the environment. The positioning error is zero after 1.6 seconds using the closed-loop control, while in open-loop control, due to the modeling error of the simulation model, the position error does not reach zero. As shown in Tab. 3.5, the average positioning error using closed-loop control is about 9.74 *mm* less than that using open-loop control.

Figure 3.8: Comparison of positioning errors between open-loop control and closed-loop control with unknown disturbance

The positioning errors in the appearance of disturbance is shown in Fig. 3.8. The closed-loop control is effective for positioning task with the positioning error being zero after 3.2 seconds and the average error being 3.23 *mm* (see Tab. 3.5). However, for open-loop control, the positioning error is larger than that without disturbance.

3.8.3.2 Tracking

The tracking performances are shown in Fig. 3.9 with the initial tension of the cables being 0.87 (N). The trajectory of the effector is defined inside the workspace of the robot and is set as a circle of radius 20 mm on the xy plane. The tracking speed is 0.2 rad/s.

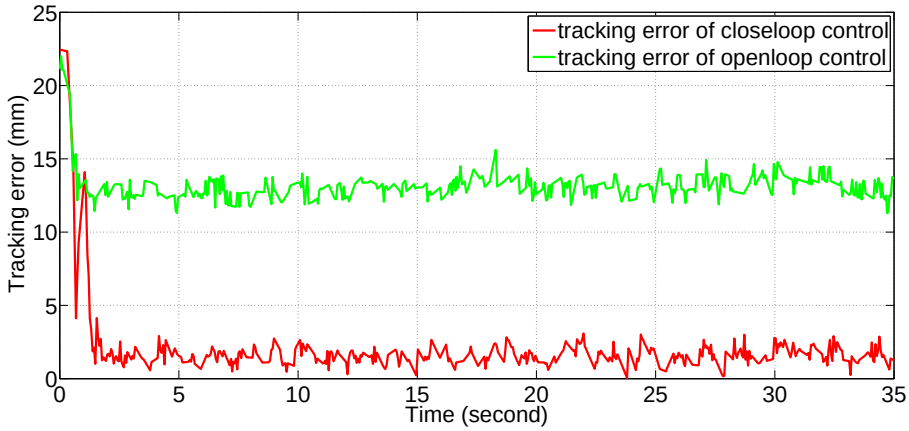

Figure 3.9: Comparison of tracking errors between open-loop control and closed-loop control

Control parameters and exact average errors are shown in Tab. 3.5. The tracking performance is much better with closed-loop control than that with open-loop control (see average errors from Tab. 3.5 : 1.94 mm for closed-loop control and 13.13 mm for open-loop control).

Table 3.5: Control parameters and average errors in all experiments

Experiments	Control parameters	Average errors (mm)
closed-loop positioning	0.12	2.46
open-loop positioning	0.5	12.20
closed-loop positioning with disturbance	0.06	3.23
open-loop positioning with disturbance	0.5	15.84
closed-loop tracking	0.12	1.94
open-loop tracking	0.5	13.13

We also test the closed-loop control for another situation where the moving object has no preassigned path. As shown in Fig. 3.10, the robot can interactively follow a moving object which position is being tracked. There is no precomputation or preassigned path needed by the control proposed in this paper. The positions of both the object and the end effector are extracted by cameras at each step. After measuring the relative difference of position between the robot effector and the moving object, the control assigns the displacement of the object to the effector position goal.

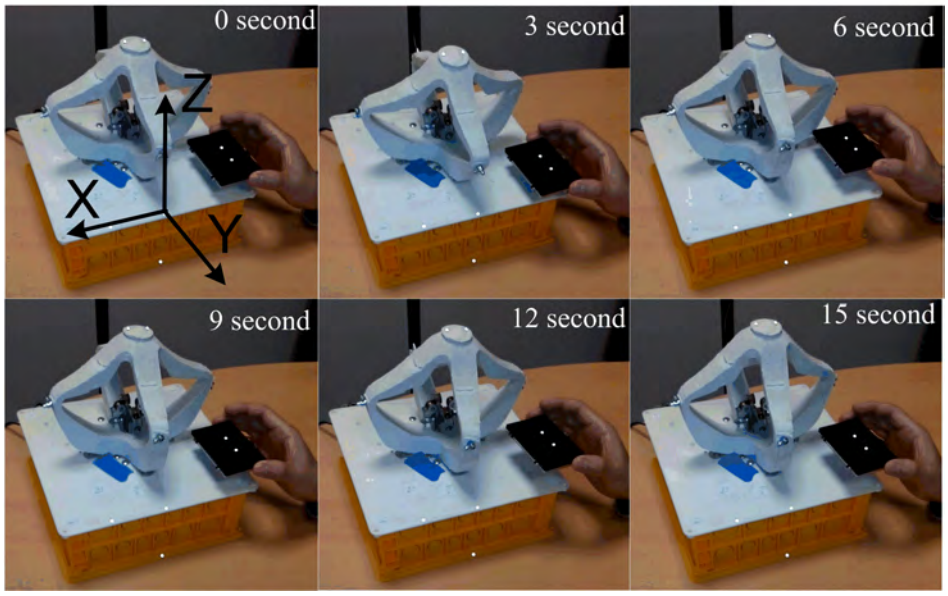

Figure 3.10: Soft robot tracks a moving object without a preassigned path. The object lies on the middle of the two markers which are placed on the black plane, and the controlled point is the end effector of the robot (middle point of two markers on the tip of the robot).

3.8.4 The Second Implementation of Closed-Loop Control

The tracking performances with the second implementation of closed-loop control are shown in this section. The desired trajectory is predefined as a circle with radius 15 *mm* and the desired angular velocity is $1/30$ (*rad/step*). In this case, the position vector $\delta_{d,k+1}$ is known for controller 1 (see Fig. 3.3). However, it is unknown for controller 2 so that we simplify the controller 2 as $u^s = -k_p^s \hat{\mathbf{J}}^+(\hat{\mathbf{x}})\, e^s$.

We express the coordinates of the robot and of the simulation model in a unique coordinate system. To get the minimum tracking error, the optimal control parameter is tuned through intensive tests. In the experiments, the control parameters are chosen as $k_p = k_p^s = 0.5$.

The 3-D trajectory of end effector within one period (191 sampling times) is illustrated in Fig. 3.11. The average tracking errors for the real robot and the simulation model are 0.34 *mm* and 0.89 *mm* respectively. The synchronous movement can be guaranteed between the simulation model and the soft robot so that the Jacobian matrix can be predicted by the simulation model for the controller design.

The control inputs of the actuators are illustrated in Fig. (3.12). Because of the periodical movement of the end effector, the control inputs are also periodical.

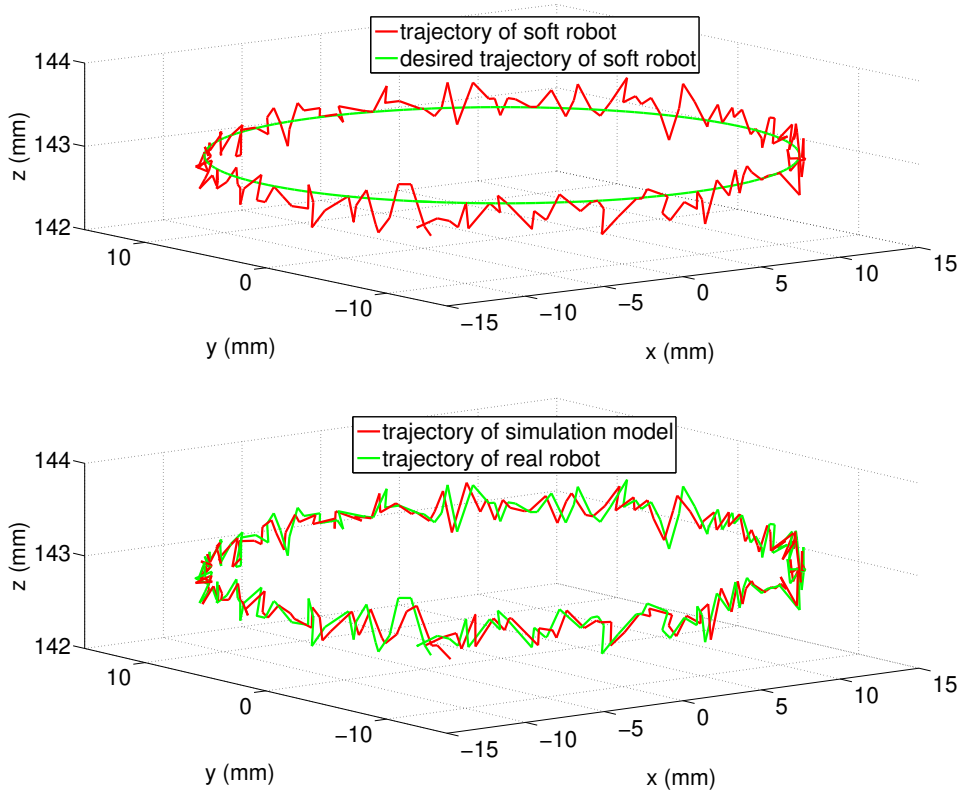

Figure 3.11: 3-D trajectories of real robot and FE model using the closed-loop controller.

3.8.5 Control Performances of Switched Control

The control performances of switched control strategy are tested on the robot which is controlled to track the desired trajectory: a circle with radius 15 *mm* and angular velocity $1/30$ (*rad/step*). When the image features are available, the closed-loop controller is activated with control parameters $k_p = k_p^s = 0.5$. The open-loop controller with $k_p^s = p = 0.5$ is employed if the image features loss is detected. In the experiments, the image feature loss occurs between sampling steps 100-120, 180-220, and 350-600. The desired trajectory of the FE model is the trajectory of real robot. However, when an image feature loss happens, we can switch to the predefined trajectory for the FE model.

The trajectory of end effector is shown in Fig. 3.13. If the interval of image feature loss is shorter, the desired trajectory can be tracked with a high accuracy. However, the tracking performance deteriorates when the interval is larger. It is noted that the performance of the proposed switch controller highly depends on the simulation model. Thanks to the high accuracy of finite element model, the tracking error is always limited in a small bound.

The control inputs are shown in Fig. 3.14. In the switching time, the chattering phe-

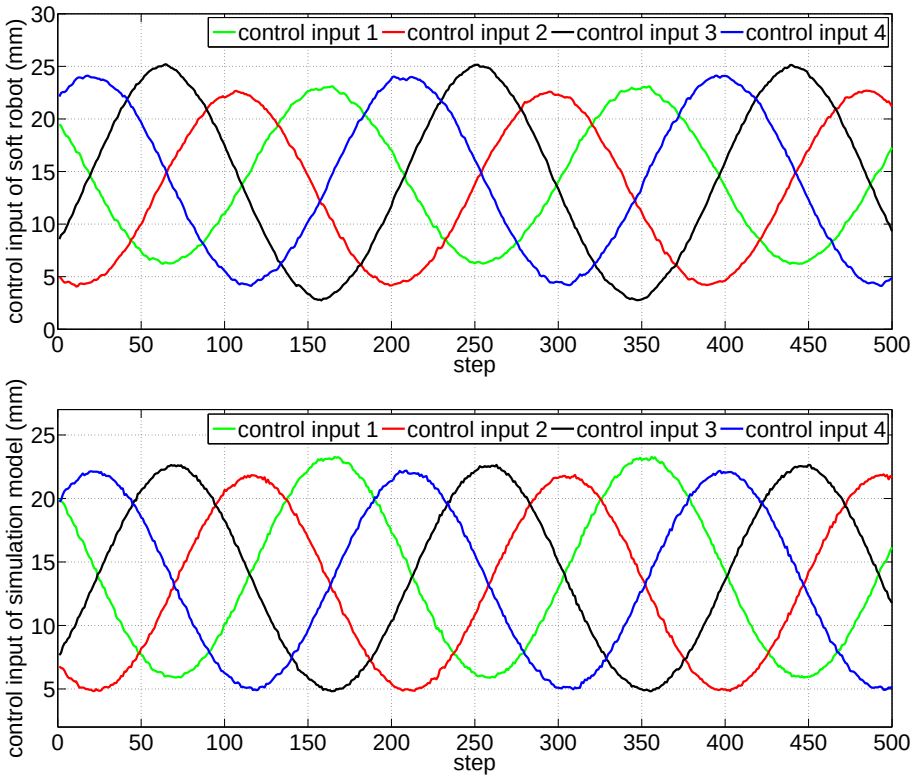

Figure 3.12: Control inputs for the soft robot and FE model using the closed-loop controller

nomenon becomes more significant. Without the feedback correction at each sampling time, the control inputs with open-loop control are smooth, however at the cost of larger tracking error.

3.9 Conclusion and Discussion

A general discrete-time kinematic model for soft robots with a wide variety of configurations is derived based on the real-time Finite Element (FE) method. The derived model provides the mathematical transformation between the actuator space and the task space directly. The Jacobian matrix of soft robots is predicted by simulating the soft robot in real-time. A vision-based calibration method is developed using two optimization methods: simultaneous optimization and sequential optimization. Based on the predicted Jacobian matrix, two closed-loop position control methods are designed with feedback signal being extracted by means of visual servoing. Considering the prediction error of Jacobian matrix, the robust stability of the closed-loop system is analyzed using Lyapunov stability theory. Besides, to deal with the problem of image feature loss, a switched control strategy is proposed by combining both the open-loop

controller and the closed-loop controller.

The performances of the closed-loop control method is affected by the inaccuracy of FE models. The main limitation of this work is the quasi-static assumption: we assume that the robot moves at low velocity. If we want to handle faster motion, we would need to design a control based on a dynamic model of the soft robot. It would allow to compensate the vibrations of the robot.

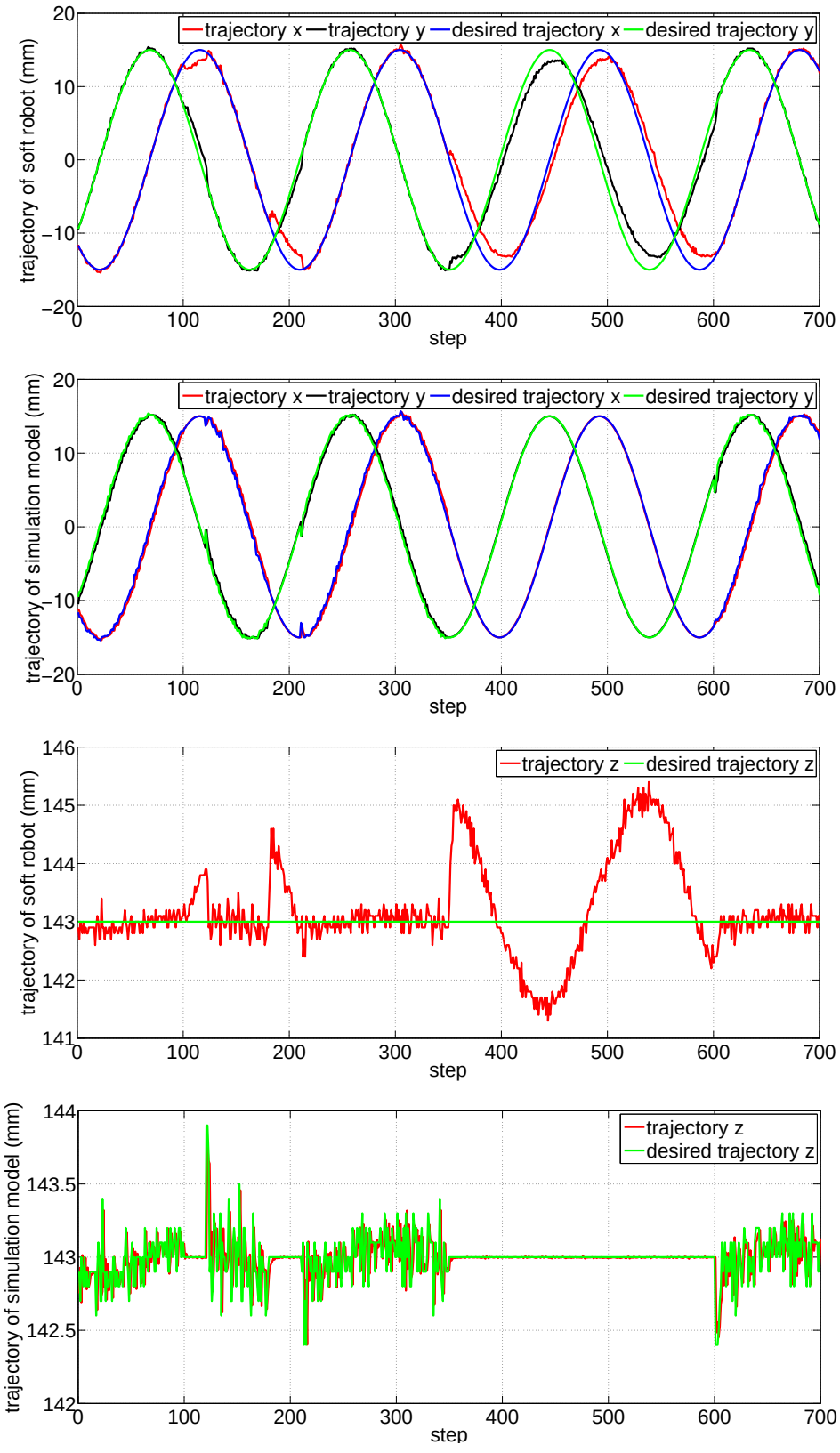

Figure 3.13: End effector trajectories of soft robot and simulation model using the switched control strategy.

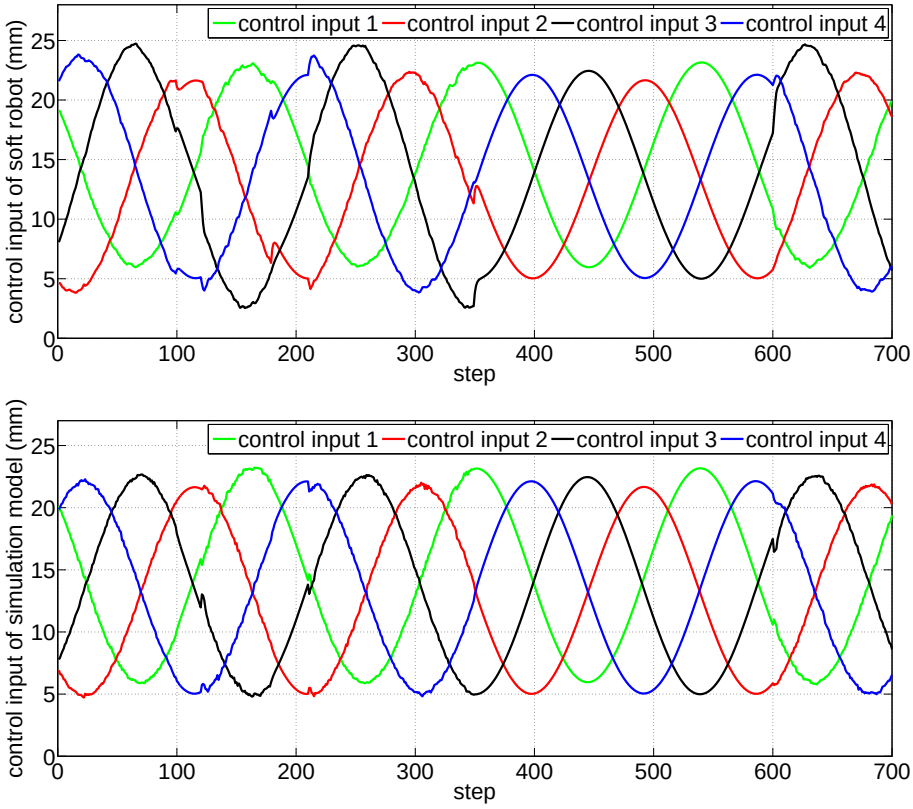

Figure 3.14: Tracking control inputs of soft robot and simulation model using the switched control strategy.

Chapter 4

Vision-based External Forces Sensing Using Finite Element Model

4.1 Overview

Benefiting from the deformability of soft robots, sensing external forces exerted on them can be made possible through visual observations provided by an external vision sensor, instead of mechatronic force sensors. The work in this chapter is based on the papers [129] and [80]. This chapter presents two possible implementations of external force sensing for soft robots and soft objects based on the real-time FE model.

For the implementation of the first strategy (marker-based), we need to put several markers on the robot surface and define the corresponding feature points on the FE model. The displacement of feature points is computed by comparing the position of markers and its corresponding feature points. By solving an optimization problem, both the intensities and the locations of external forces can be estimated.

The second implementation (marker-free) uses point cloud data acquired by an RGB-D camera and consists of two steps: location detection and intensity computation. The locations are estimated based on the segmentation of the point cloud. Unlike the marker-based approach, we predefine feature points on the surface of FE model and find their corresponding feature points from the point cloud by the registration along the normal direction of the surface. Then, the intensities can be computed by solving an inverse quasi-static problem.

This chapter is organized as follows. The model of soft robots with external forces is introduced in Section 2. Section 3 gives an overview of the implementation of external force sensing strategy. Then, the marker-based and marker-free external force sensing methods are presented respectively in Section 4 and Section 5. The validation by both the simulation

and experiment for marker-based approach is shown in Section 6 and Section 7, respectively. Section 8 shows the experiment results of marker-free external force sensing. Finally, the conclusion and discussion are presented in Section 9.

4.2 Modeling of Soft Robots with External Forces

Using FE method, the nodes can be divided into two groups: actuators (the nodes where the external inputs act on) and effectors (feature points). Position vectors of actuators and effectors are defined as δ_a and δ_e, respectively. Considering the external forces, the quasi-static equations of soft robots can be written as (see equations (2.4) and (2.5)):

$$\delta_{e,k+1} = \mathbf{W}_{ea}(\mathbf{x}_k)\lambda_{a,k} + \mathbf{W}_{ef}(\mathbf{x}_k)\lambda_{f,k} + \delta_{e,k}^{free} \tag{4.1}$$

$$\delta_{a,k+1} = \mathbf{W}_{aa}(\mathbf{x}_k)\lambda_{a,k} + \mathbf{W}_{af}(\mathbf{x}_k)\lambda_{f,k} + \delta_{a,k}^{free} \tag{4.2}$$

where $\mathbf{W}_{ij}(\mathbf{x}_k) = \mathbf{H}_i(\mathbf{x}_k)\mathbf{K}^{-1}(\mathbf{x}_k)\mathbf{H}_j^T(\mathbf{x}_k)$ ($i = e, a$, and $j = a, f$). $\mathbf{H}_f^T\lambda_f$ gathers the contribution of the external forces (where $\mathbf{H}_f$ is the mapping matrix between nodes and external force directions and λ_f is the contribution vector of external forces). δ_e^{free} and δ_a^{free} are respectively position vectors of effectors and actuators when $\lambda_a = \lambda_f = 0$. Using $\mathbf{W}_{ea}(\mathbf{x})$, we can get a measure of the mechanical coupling between effectors and actuators. $\mathbf{W}_{ef}(\mathbf{x})$ measures the coupling between effectors and external forces. The mechanical coupling between actuators and between actuators and external forces are represented by matrices $\mathbf{W}_{aa}(\mathbf{x})$ and $\mathbf{W}_{af}(\mathbf{x})$, respectively.

With the continuity assumption of $\mathbf{W}_{ea}(\mathbf{x}_{k-1}) \approx \mathbf{W}_{ea}(\mathbf{x}_k)$ and $\mathbf{W}_{ef}(\mathbf{x}_{k-1}) \approx \mathbf{W}_{ef}(\mathbf{x}_k)$, (4.1) can be simplified as:

$$\delta_{e,k+1} = \delta_{e,k} + \mathbf{W}_{ea}(\mathbf{x}_k)\triangle\lambda_{a,k} + \mathbf{W}_{ef}(\mathbf{x}_k)\triangle\lambda_{f,k} \tag{4.3}$$

where $\triangle\lambda$ is the incremental vector of λ.

We assume that the number of effectors, actuators and external forces are respectively n_e, n_a and n_f. If there is no linear dependency between lines of $\mathbf{H}_a$, $\mathbf{H}_e$ and $\mathbf{H}_f$, the features of the the matrices $\mathbf{W}_{aa}(\mathbf{x})$, $\mathbf{W}_{ea}(\mathbf{x})$ and $\mathbf{W}_{ef}(\mathbf{x})$ are:

(1) $\mathbf{W}_{aa}(\mathbf{x})$ is always positive-definite.

(2) $\mathbf{W}_{ea}(\mathbf{x})$ and $\mathbf{W}_{ef}(\mathbf{x})$ have full row rank when $n_e \leq n_a$ and $n_e \leq n_f$. In the opposite case, $\mathbf{W}_{ea}(\mathbf{x})$ and $\mathbf{W}_{ef}(\mathbf{x})$ are matrices of full column rank.

4.3 Overview of Implementation

In this section, we introduce the implementation (see Fig. 4.1) to estimate external forces for soft robots.

The implementation requires a FE model of soft robot and a tracking system. Based on the constitutive law of the material (in this study, we have considered a hookean material), the FE model provides the compliance matrix between the external forces and the tracked points (called "effectors"). The tracking system is used to get the 3D position of these points which are defined in both the soft robot and its FE model. Based on the position error of effectors on soft robot and its FE model, the external force can be computed at each sampling time.

The computed external force λ_f, together with the control input λ_a, contributes to update the configuration of the FE model. It is noted that control inputs for the soft robot and its FE model are same so that their influences on both systems can be compensated. However, to compute the incremental external forces within each step, the coupling between the external forces and control inputs should be considered.

As shown in Fig. 4.1, the actuator input λ_a is sent to both the soft robot and its FE model. When the external force acts on the soft robot, there is a deflection between the FE model and the real robot. The external forces λ_f are computed by the External Force Computation (EFC) algorithm so that the feature points on the FE model δ_e can track their corresponding points on the real robot δ_e^r.

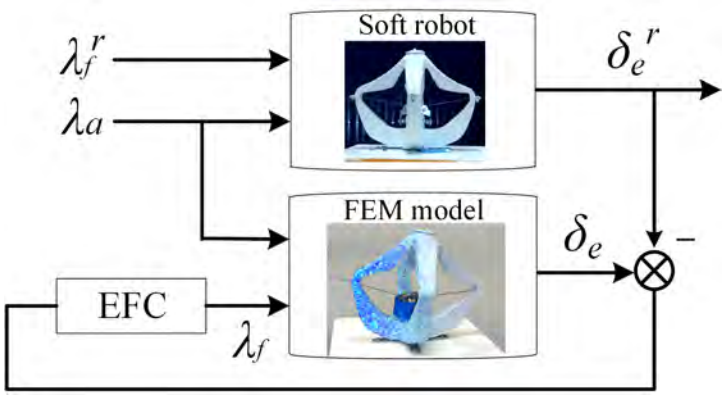

Figure 4.1: Framework of external force sensing for soft robots. EFC (External Force Computation) module is the algorithm to compute external forces. λ_a is the actuator input for both soft robot and its FE model. λ_f^r and δ_e^r are respectively are the external force and position of effectors for the soft robot. λ_f and δ_e are the corresponding variables for the FE model.

The advantages of this strategy are: (1) it can be employed to compute external forces acting on soft robots with complicated structures; (2) The implementation does not need to track the entire shape of the robot; (3) We rely on the input values (from the controller) of the

actuator so we don't need actuator force sensors (but we could integrate this information for more precise results); (4) The modeling is based on an elastic FE model without assumption on the constitutive law of the material.

4.4 Marker-based External Force Sensing

In this section, the sufficient condition to realize the marker-based external force sensing is investigated and the forces are computed by solving an inverse problem using an optimization-based strategy.

4.4.1 Algorithm of External Force Computation

The task for external force computation can be converted to compute $\triangle\lambda_f$ so that δ_e can track its desired value δ_e^r. In the following, we propose an optimization-based method to compute the external forces. For the implementation, we assume that the locations of external forces are known so that we can compute the matrices $\mathbf{W}_{ea}$ and $\mathbf{W}_{ef}$.

At each sampling time, the coupling between the external forces and actuators is considered for the computation. We rewrite the equation of the FE model as

$$\delta_{e,k+1} = \delta_{e,k} + \mathbf{W}_{eI}(\mathbf{x}_k)\triangle\lambda_{I,k} \tag{4.4}$$

where $\mathbf{W}_{eI} = \begin{bmatrix} \mathbf{W}_{ea} & \mathbf{W}_{ef} \end{bmatrix}$ and $\triangle\lambda_I = \begin{bmatrix} \triangle\lambda_a & \triangle\lambda_f \end{bmatrix}^T$.

To get the contribution of external forces, the optimization problem can be defined as $\triangle\lambda_{f,k} = \arg\min\Gamma\left(\triangle\lambda_{I,k}\right)$ where $\Gamma\left(\triangle\lambda_{I,k}\right)$ is the objective function:

$$\Gamma\left(\triangle\lambda_{I,k}\right) = \frac{1}{2} \parallel \mathbf{W}_{eI}(\mathbf{x}_k)\triangle\lambda_{I,k} - \left(\delta_{e,k} - \delta_{e,k}^r\right) \parallel^2 \tag{4.5}$$

In the case of constrained optimization, the numerical optimization algorithm (like active set method) is necessary to be employed. In order to match the standard quadratic programming (QP) formulation for the usage of software packages [130], (4.5) can be converted to be

$$\min_{\triangle\lambda_{a,k},\triangle\lambda_{f,k}} \frac{1}{2}\left(\triangle\lambda_{I,k}\right)^T \mathbf{Q}_k\triangle\lambda_{I,k} + \mathbf{c}_k^T\triangle\lambda_{I,k} \tag{4.6}$$

$$\text{s.t.} \quad \triangle\lambda_{a,k} = \triangle\lambda_{a,k}^r \tag{4.7}$$

$$\text{or} \quad \triangle\delta_{a,k} = \triangle\delta_{a,k}^r \tag{4.8}$$

where $\mathbf{Q}_k = \mathbf{W}_{eI}^T(\mathbf{x}_k)\,\mathbf{W}_{eI}(\mathbf{x}_k)$ and $\mathbf{c}_k^T = -\left(\delta_{e,k} - \delta_{e,k}^r\right)^T \mathbf{W}_{eI}$. $\triangle\lambda_{a,k}^r$ or $\triangle\delta_{a,k}^r$ are the control inputs generated by the motion controller. For robots actuated by air pressure, $\triangle\lambda_{a,k}$ is usually chosen as the control input. For robots actuated by cables, $\triangle\delta_{a,k}$ can be chosen as control input and can also be converted into the force constraint of $\triangle\lambda_{a,k}$.

For soft objects without actuators and other constraints, (4.5) has an analytical solution which can be obtained by

$$\triangle\lambda_{f,k} = \left(\mathbf{W}_{ef}^T\mathbf{W}_{ef}\right)^{-1}\mathbf{W}_{ef}^T\left(\delta_{e,k} - \delta_e^r\right) \tag{4.9}$$

It is necessary to figure out how many markers are needed for the computation of external forces. The sufficient condition to compute the force intensity is shown as the following corollary:

Corollary 1: *Given the locations of external forces, the intensity can be computed if $m \geq n$ where m and n are respectively the number of rows and columns in the compliance matrix $\mathbf{W}_{ef}$.*

Proof: For the implementation of strategy shown in Fig. 4.1, we have $\triangle\lambda_{a,k} = \triangle\lambda_{a,k}^r$ where the superscript r represents the corresponding variables for real robot. We assume that the FE model is accurate so that we have $\mathbf{W}_{ea}^r(\mathbf{x}_k) = \mathbf{W}_{ea}(\mathbf{x}_k)$ and $\mathbf{W}_{ef}^r(\mathbf{x}_k) = \mathbf{W}_{ef}(\mathbf{x}_k)$.

Without considering the computation delay of the implementation, from (4.3), we have

$$\triangle\delta_{e,k}^r - \triangle\delta_{e,k} = \mathbf{W}_{ef}(\mathbf{x}_k)\left(\triangle\lambda_{f,k}^r - \triangle\lambda_{f,k}\right) \tag{4.10}$$

where $\mathbf{W}_{ef} \in \mathbb{R}^{m\times n}$ has full row rank ($m \geq n$) or full column rank ($m \leq n$).

In order to make sure that $\triangle\lambda_{f,k}^r - \triangle\lambda_{f,k}$ have unique zero solution when $\triangle\delta_{e,k}^r = \triangle\delta_{e,k}$, we need $m \geq n$.

It is easy to make the sufficient condition $m \geq n$ hold by choosing more feature points as effectors. It is noted that m and n correspond to the number of feature points and that of the external forces, respectively. Usually, more feature points are employed so that the accurate forces can be computed even if some feature points are hidden at some sampling time. In this case, the compliance matrix $\mathbf{W}_{ef}$ and the vector $\left(\delta_{e,k} - \delta_e^r\right)$ should be recomputed by removing the corresponding rows.

Remark 1: It is noted that the constraints in QP formulation (4.6) can be deleted to compute both the external force and control input. This strategy can reduce the delay generated by the measure of feature points or control inputs. However, we needs more feature points to make sure that the optimization has unique solution. In addition, if the FE model is not accurate, the increased number of optimization variables could also reduce the accuracy of force computation. With this in mind, we employ the QP formulation with the constraints.

4.4.2 Estimation of External Forces Location

In this section, we first prove a property: the locations of external forces can be estimated just based on the displacement of feature points. Then, we propose a simple strategy to estimate the locations.

In order to choose the number of feature points for the location estimation of external forces, the following corollary is needed.

Corollary 2: *Given the number of possible locations n_p, the locations of external forces can be estimated if the minimal number of feature points is n_p.*

Proof: All the nodes (the number is n_{all}) of the FE model are numbered so that we can get the locations if we get the corresponding indexes of actuated nodes. S_p is the number set of possible locations. We assume that the actual external forces λ_1 are applied on the node set S_1 ($S_1 \subseteq S_p$) and the displacement of defined effectors is $\triangle\delta_e$. Then, λ_1 can be written in the argumented form λ_1^a with n_{all} elements and the zero elements means that there is no force applied on the corresponding nodes. From (4.4), we have

$$\triangle\delta_e = \mathbf{W}_{eI}^a \triangle\lambda_1^a \tag{4.11}$$

where $\mathbf{W}_{eI}^a$ is the compliance matrix between the defined effectors and the argumented force vector.

If there is other group of external forces (the argumented vector λ_2^a) applied on the node set S_2 ($S_1 \cup S_2 \subseteq S_p$) to generate the same displacement of defined effectors $\triangle\delta_e$, (4.11) can be written as

$$\triangle\delta_e = \mathbf{W}_{eI}^a \triangle\lambda_2^a \tag{4.12}$$

Combining both (4.11) and (4.12), we have

$$\mathbf{W}_{eI}^a (\triangle\lambda_1^a - \triangle\lambda_2^a) = 0 \tag{4.13}$$

We assume that the number of possible locations is n_p. Considering that $S_1 \cup S_2 \subseteq S_p$, the largest number of non-zero elements in vector $(\triangle\lambda_1^a - \triangle\lambda_2^a)$ is n_p. If (4.13) has unique zero solution, i.e. $\triangle\lambda_1^a = \triangle\lambda_2^a$, the location and intensity of external forces can be estimated. We assume that the number of elements in $\triangle\delta_e$ is n_e which is equal to that of the rows in $\mathbf{W}_{eI}^a$. If $n_e \geq n_p$, the unique solution of (4.13) is $\triangle\lambda_1^a = \triangle\lambda_2^a$.

Therefore, with unknown external forces in terms of locations and intensity, they can be uniquely determined if the number of defined effectors n_e is not less than that of the candidate external forces n_p.

Based on this corollary, we show our strategy into two steps to estimate the locations.

At the first step, the possible locations of external forces (the number is n_p) are predefined. We also need to define more effectors (the number is n_e and $n_e \geq n_p$) to realize the estimation.

Then, the intensities of all candidate external forces are computed at the same time using the optimization (4.6). The non-zero elements in $\triangle\lambda_{I,k}$ correspond to the indexes of external forces.

Remark 2: For real applications, the FE model is not precise so that the zero elements in the idea case maybe become non-zero. However, if the error of FE model is limited in a small bound, their absolute values are still much smaller than other elements. The absolute vector of all external forces is obtained by accumulating the computation result of each step. By setting a threshold for the elements in the vector, the indexes of the elements which absolute values are larger than the threshold are the correct locations.

4.5 Marker-free External Force Sensing

The objective of this section is to describe the method to sense the external forces based on the FE model and using an RGB-D camera. The force sensing method can be separated into two parts: location recognition based on the segmentation of the point clouds provided by the camera, and intensity computation based on the deflection of selected feature points.

4.5.1 Location Recognition and Feature Selection

As shown in Fig. 4.2, the image processing method consists of five steps: point cloud filter, segmentation, location recognition, location mapping and feature point selection. In the following, we introduce the implementation of each step.

4.5.1.1 Point Cloud Filter

The surfaces of both the soft robot and its surrounding (the manipulation tools for instance) are captured by an RGB-D camera through successive 3D point clouds. The acquired point cloud is then filtered through the following process. We employ the PassThrough, VoxelGrid filters and RadiusOutlier removal to remove points whose values fall outside a given interval along a specified dimension, to downsample the point cloud and to remove outliers from noisy data, respectively [131].

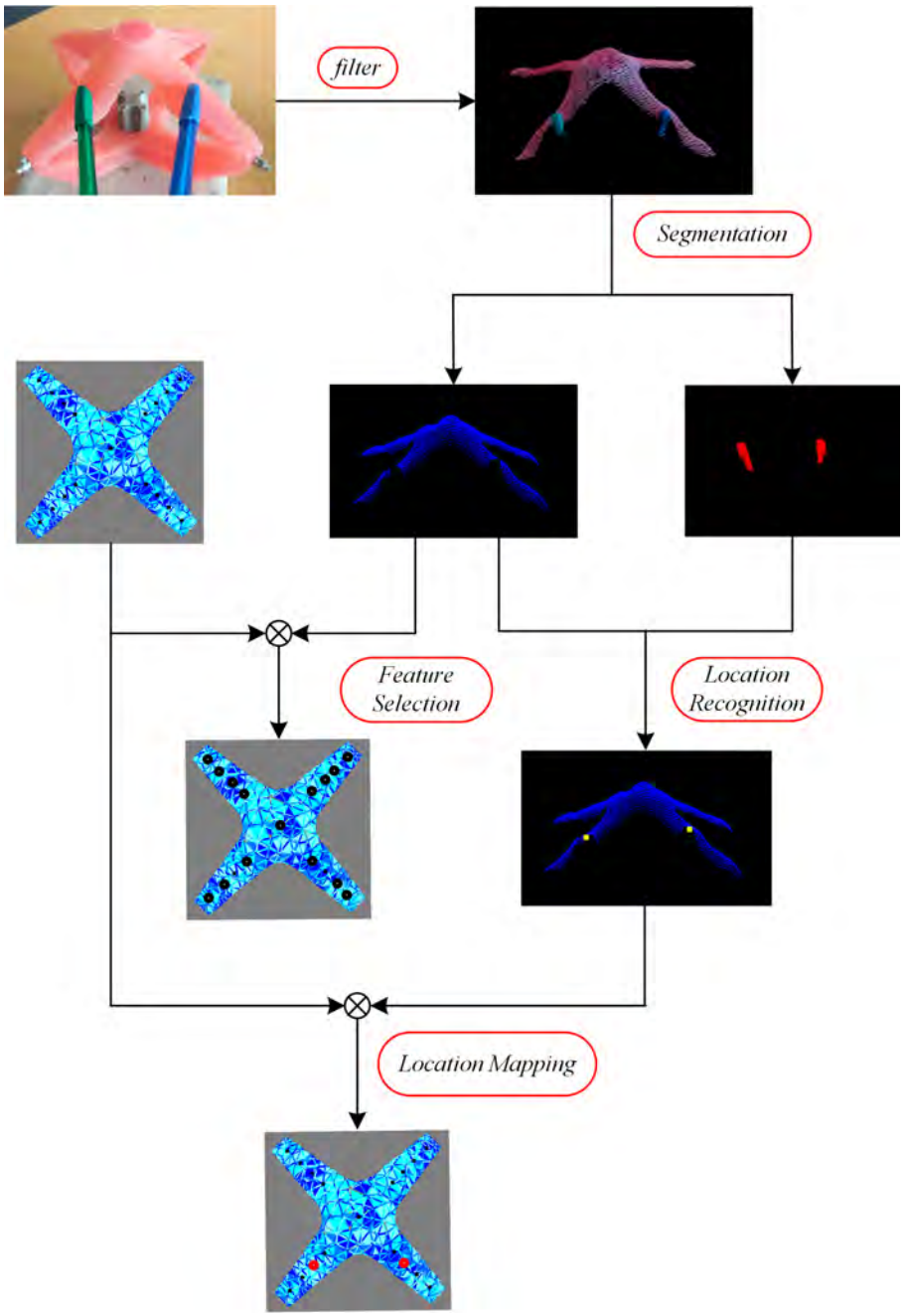

Figure 4.2: Location recognition and feature selection. The image processing method consists of five steps: filter, segmentation, location recognition, location mapping and feature point selection.

4.5.1.2 Segmentation

We assume that the color of the robot is homogeneous and is different with respect to the surrounding objects. Using a color-based region growing segmentation technique [131], the

filtered point cloud is segmented into two parts: the point cloud of the soft robot (shown in blue in Fig. 4.2) and the point cloud of area (shown in red in Fig. 4.2).

4.5.1.3 Location Recognition

In this component, the contact location is estimated based on the segmented contact area and robot point clouds. The algorithm is summarized in Algorithm 4.1.

Algorithm 4.1 Location recognition

1. **Define** a constant search radius r_{search} and
   a threshold number of neighbors n_{neigh}
2. **If** objects are found after segmentation
   For each point in contact point cloud
   Search neighbors (r_{search}) in robot point cloud
   If the number of neighbors is larger than n_{neigh}
   Feature points detected
   End
   End
   Separate the detected points into groups
   For each group
   Compute the average position
   End

End

When the objects around the robot are found after segmentation, the location recognition algorithm starts to find the contact points between the objects and the robot. The contact points are usually difficult to be detected because they can be occluded by the manipulation operator. When the contact between the soft robot and its surrounding environment occurs, some points in the point cloud corresponding to the object in contact get close to the robot point cloud. We set a constant search radius r_{search} around the point among the contact point cloud to detect if there is a contact. Then, a KdTree [131] is employed for each point in the contact point cloud to find its neighbors within the specified radius in the robot point cloud. If the number of neighbors is larger than a defined threshold number n_{neigh}, we regard this point as a contact point.

When all the contact points are collected, they are clustered into several groups based on the strategy of euclidean cluster extraction. By using the distance information of point cloud, clustering does not depend on color information. Then, the average position of points in each group is computed as the location of a external force.

4.5.1.4 Location Mapping

Using the method described above, the location of external forces can be detected based on the segmentation of point cloud. Then, the detected location is mapped to the surface of FE model where multiple locations are predefined and are distributed on the surface. For each detected location, we find the closest location on the FE model to the detected one so that the external forces can be applied on the FE model.

4.5.1.5 Feature Point Selection

As shown in Fig. 4.1, external force sensing requires the computation of the displacements of feature points defined on the surface of both the soft robot and its FE model. Using the marker-based force sensing [129], the feature points on the soft robot can be obtained by putting several markers on its surface. If there are no obvious feature points on the surface of the soft robot, the computation of the displacement is not available using the marker-based strategy. To deal with this problem, the feature point selection is implemented by registering the FE model on the robot point cloud. The algorithm is summarized in Algorithm 4.2.

Algorithm 4.2 Feature point selection

1. **Define** feature points on the surface of FE model and
   two threshold distance d_m and d_l
2. **For** each feature point $\mathbf{x}_f$
   Compute the normal direction
   For each point in robot point cloud
   Compute distance between $\mathbf{x}_f$ to normal line
   End
   Compute the minimum distance d_{mini}
   If $d_{mini} > d_m$
   The feature point is hidden
   Else possible registered point $\mathbf{x}_p$ obtained
   End
   Compute distance d between $\mathbf{x}_p$ to $\mathbf{x}_f$
   If $d < d_l$
   registered point $\mathbf{x}_c$ obtained
   End

End

In order to capture the deformation of robot, the FE model of the soft robot is built along with multiple predefined feature points, evenly distributed on the surface. Then, the predefined feature points are gathered together as the feature point cloud which can be employed to approximate the surface of robot. The registration of feature points (see Fig. 4.3) is implemented

along the normal direction which is computed using the predefined feature points on the surface of FE model. The NormalEstimation class in point cloud library is employed to compute the normal direction for each feature point. For each feature point, we find its registered point in robot point cloud which has the minimum distance d_{mini} to the normal line. Each feature point and its registered point are regarded as a pair of corresponding feature points so that the displacement can be computed.

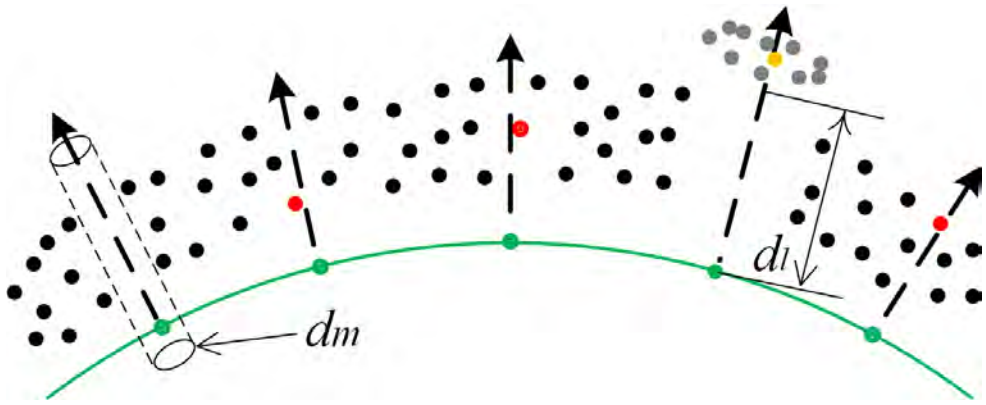

Figure 4.3: Registration of feature points along the normal direction. The black, green, gray, orange and red points are respectively the robot point cloud, feature point cloud, outliers after segmentation, registered points in the outliers and the correct registered points. d_m and d_l are the predefined threshold values to get rid of imprecise registered points.

For real applications, some feature points may not be available because of occlusions. During the registration of the feature points, the hidden feature points on the FE model need to be detected. Consequently, we predefine a threshold distance d_m to discard potential outlier feature point. If the distance is larger than the threshold, the feature point is not available and its displacement will not be considered for the intensity computation introduced in the next subsection.

The registration process should also avoid the situation for which the point cloud is not well segmented, e.g. objects with colors similar to the ones of the robot. A threshold distance d_l is predefined to detect the registered points which falls into this situation. If the distance between a feature point and its registered point is larger than d_l, the registered points are considered as an outlier and should be deleted too.

4.5.2 Intensity Computation

It is indeed easier to capture the normal direction than the other directions. For the same number of feature points on the FE model, this strategy can dramatically reduce the constraint

size and further reduce the computation time. For the intensity computation, it is necessary to map the quasi-static equation (4.1) to the motion space along the normal of the robot surface:

$$\delta_e^N = \mathbf{W}_{ea}^N \lambda_a + \mathbf{W}_{ef}^N \lambda_f + \delta_e^{free,N} \tag{4.14}$$

Given the locations of contacts, the compliance matrices $\mathbf{W}_{ea}^N$ and $\mathbf{W}_{ef}^N$ can be obtained using the FE method. Based on the method described in Fig. 4.1, the intensity of the external forces on the contact points can be computed by solving an optimization problem $\lambda = \arg\min \Gamma(\lambda)$ where $\Gamma(\lambda)$ is the objective function:

$$\Gamma(\lambda_k) = \frac{1}{2} \parallel \mathbf{W}_{eI}^N(\mathbf{x}) \lambda_I + \delta_e^{free,N} - \delta_d^N \parallel^2 \tag{4.15}$$

where $\mathbf{W}_{eI}^N = \begin{bmatrix} \mathbf{W}_{ea}^N & \mathbf{W}_{ef}^N \end{bmatrix}$ and $\lambda_I = \begin{bmatrix} \lambda_a & \lambda_f \end{bmatrix}$. δ_d^N is the position of the registered points along the normal direction. The computed external forces λ_f and λ_a are employed to update the FE model to reach the same configuration of soft robot observed by the camera.

(4.15) can be converted to a standard quadratic programming (QP) formulation:

$$\min_{\lambda_k} \frac{1}{2} \lambda^T \mathbf{Q} \lambda + \mathbf{c}^T \lambda \tag{4.16}$$

$$\text{s.t.} \quad \lambda_a = \lambda_a^r \tag{4.17}$$

$$\text{or} \quad \begin{bmatrix} \mathbf{W}_{aa}^N & \mathbf{W}_{af}^N \end{bmatrix} \lambda + \delta_a^{free,N} = \delta_a^{r,N} \tag{4.18}$$

where $\mathbf{Q} = \left(\mathbf{W}_{eI}^N\right)^T \mathbf{W}_{eI}$ and $\mathbf{c} = \mathbf{W}_{eI}^N \left(\delta_e^{free,N} - \delta_d^N\right)$. λ_a^r and δ_a^r are respectively the actuator force and the actuator displacement for both the soft robot and its FE model. The control input are considered for the computation by constraints (4.17) and (4.18). (4.17) and (4.18) are employed respectively when the actuator force and the actuator displacement are chosen as the control input.

Remark 3: The dimension of $\mathbf{Q}$ is equal to the number of external forces and actuators so that the computation cost for QP optimization (4.16) remains almost constant even if a larger number of feature points are employed for the computation.

4.6 Validation of Marker-based Approach by Simulation

A first verification of our algorithm has been performed using simulation. It allows to verify the precision we can expect from our algorithm by setting aside, temporarily, the problems of camera sensors inaccuracies. For the implementation, two FE models are needed: one

for direct simulation and the other for inverse simulation. Using this setup, we can easily define the external forces, test the sensitivity to the parameters of the soft robot model, and benchmark the algorithm for different control inputs.

Two types of structures are tested in this section: a soft sheet and a soft parallel robot. The soft sheet is passive: there is no actuator placed on its structure. However, we consider the control inputs for the soft parallel robot (for each actuator, the force is defined as $2N$.). In order to evaluate the sensibility, we also consider the case when the inverse model is not accurate. The setup of both cases consists of a direct FE model and an inverse FE model (see Fig. 4.4). The Young's Modulus for both direct FE models of soft sheet and soft robot are defined as 300 kPa. The elements number and sampling frequency are respectively 4923 and 16Hz for the soft sheet, and 4147 and 20Hz for the soft robot. The same external forces $(0,0,0.4)N$ and $(0,0,-0.4)N$ are applied on both systems. Several candidate locations and forces should be defined for location estimation. Using the strategy proposed in Section 5.4, the correct locations can be determined.

4.6.1 Evaluation Index

In order to quantify the performance of the proposed strategies, we define two evaluation indexes I_e and I_w. I_e is used to evaluate the computation error and I_w is employed to evaluate the distribution of markers and candidate locations.

We want to evaluate the influence of the model inaccuracies on the estimation error of the force. Based on the computation error, the index I_e is defined as $I_e = \| \mathbf{f}^c - \mathbf{f}^a \|$ where $\mathbf{f}^c$ and $\mathbf{f}^a$ are the vector of all computed and applied force respectively.[1] Given the same applied forces, a larger I_e means a larger estimation error.

If the deformation of structures is small, the compliance matrix $\mathbf{W}_{ef}$ is close to a constant matrix so that the kinematic equation can be simplified as $\triangle\delta_e = \mathbf{W}_{ef}\triangle\lambda_f$ where $\mathbf{W}_{ef}$ can be computed at the initial configuration. Then, we have $\| \triangle\lambda_f \| \leq \| \mathbf{W}_{ef}^{+} \| \| \triangle\delta_e \|$. For the matrix $\mathbf{W}_{ef} \in \mathbb{C}^{m\times n}$ $(m \geq n)$, the evaluation index I_w is defined as $I_w = \| \mathbf{W}_{ef}^{+} \|$. If I_w is small, $\mathbf{W}_{ef}$ is well-conditioned; if I_w is large, $\mathbf{W}_{ef}$ is ill-conditioned. A smaller I_w corresponds to a better distribution of markers and candidate locations. Given the displacement error of makers, I_w can be employed to estimate the maximum error of computed forces.

[1] In the computation, each external force consists of three components, i.e. $\mathbf{f}_i = (f_x, f_y, f_z)^T$ defined in the same coordinate system. The symbol $\| * \|$ means the norm of $*$. In this section, we compute the 1-norm for all vectors and matrices.

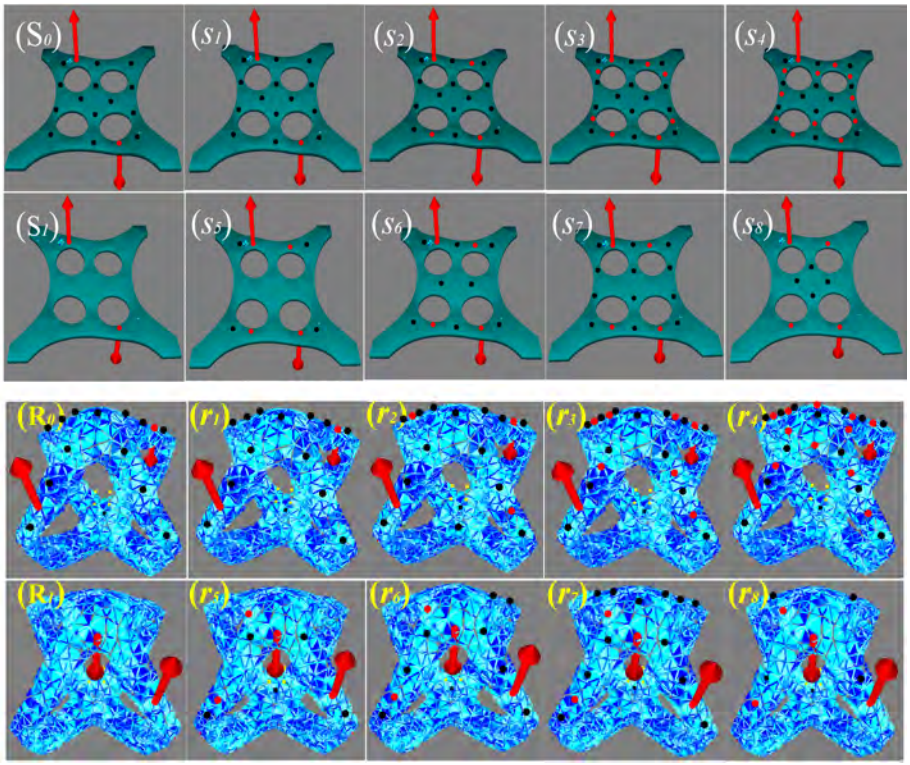

Figure 4.4: Validation setup of a soft sheet (S) and a soft parallel robot (R). The red arrows stand for the applied forces and the computed forces respectively for direct FE models and inverse FE models. The black points and the red ones show the distribution of markers and candidate locations, respectively. $\mathbf{S}_0$, $\mathbf{S}_1$ and $\mathbf{R}_0$, $\mathbf{R}_1$ are respectively the direct FE models for the soft sheet and the soft robot. $s_1 \sim s_8$ and $r_1 \sim r_8$ are inverse FE models and the differences between them are the distribution of markers and candidate locations. The same distribution of markers is defined for the groups $\mathbf{S}_0$, $s_1 \sim s_4$ and $\mathbf{R}_0$, $r_1 \sim r_4$. For the groups $\mathbf{S}_1$, $s_5 \sim s_8$ and $\mathbf{R}_1$, $r_5 \sim r_8$, the same distribution of candidate locations are employed to test the algorithm.

4.6.2 Validation Results based on Accurate Inverse Model

In this section, eight cases (two groups) are tested for each system to evaluate our algorithm. As shown in Tab. 4.1, the evaluation index I_e is limited into a very small range so that the computed forces can track the applied forces in a high accuracy. I_w is computed based on the initial compliance matrix for each case. For different distribution of markers and locations, I_w is very different. The computation error I_e could be very large if we choose the distribution with a large I_w, even if the model has a high accuracy (see the last column in Tab. 4.1).

Table 4.1: Evaluation indexes I_e and I_w for both validations. The units of I_e and I_w are mN and N/m, respectively. $s_1 \cdots s_8$ are different arrangements of marker points and candidate locations (see in Fig. 4.4).

Soft sheet	Direct FE model	$\mathbf{S}_0$				$\mathbf{S}_1$			
	Inverse FE model	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8
	I_e	0.3	1.1	4.6	33.0	2.8	1.3	1.4	130.3
	I_w	0.5	1.4	3.3	20.4	1.3	1.6	1.5	1991
Soft robot	Direct FE model	$\mathbf{R}_0$				$\mathbf{R}_1$			
	Inverse FE model	r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8
	I_e	3.2	3.2	8.7	60.2	9.0	25.2	8.8	79.9
	I_w	0.2	0.8	8.1	168.9	28.5	12.1	4.9	52

4.6.3 Error Analysis

To evaluate the algorithm with different model error, we employ the setup shown in Fig. 4.4. The errors are generated by setting different parameters or position vectors for the inverse FE model. The ground truth is the results obtained by the direct FEM with accurate parameters. We consider the imprecise Young's Modulus and the imprecise markers position as two kinds of errors for both systems. Besides, the error of actuator input for soft robot is also considered. For both systems, three imprecise data of Young's Modulus for the inverse FE model are tested: 280 kPa, 260 kPa and 240 kPa. The Young's Modulus is defined as 300kPa for both direct FE models of soft sheet and soft robot. Three errors in the position of the marker points are also used: errors of $0.05mm$, $0.1mm$ and $0.15mm$ along the z axis. Three input errors of each actuator $0.1N$, $0.15N$, and $0.2N$ are tested for soft robot.

Table 4.2: Evaluation index I_e (error of computed force in mN) for soft sheet. The reference errors are shown in Tab. 4.1. Two cases are considered: imprecise Young's Modulus and imprecise markers position. $s_2 \cdots s_7$ are different arrangements of marker points and candidate locations (see in Fig. 4.4)

Direct FE model		$\mathbf{S}_0$			$\mathbf{S}_1$		
Inverse FE model		s_2	s_3	s_4	s_5	s_6	s_7
Imprecise Young's Modulus	280	109.8	197.9	278.3	137.7	118.4	114.0
	260	216.3	395.4	551.1	273.5	234.7	224.9
	240	319.4	590.2	807.9	404.4	347.8	332.4
Imprecise markers position	0.05	132.6	145.4	307.7	57.7	131.2	131.1
	0.10	265.1	290.1	612.0	113.4	262.5	262.2
	0.15	397.5	434.8	918.2	169.1	393.9	393.2

For location estimation, several candidate locations are predefined for both soft sheet and soft parallel robot. The forces on all candidate locations are computed at the same time by

Table 4.3: Evaluation index I_e (error of computed force in mN) for soft robot. The reference errors are shown in Tab. 4.1. Three cases are considered: imprecise Young's Modulus, imprecise markers position and imprecise actuator input. $r_2 \cdots r_7$ are different arrangements of marker points and candidate locations (see in Fig. 4.4)

Direct FE model		$\mathbf{R}_0$			$\mathbf{R}_1$		
Inverse FE model		r_2	r_3	r_4	r_5	r_6	r_7
Imprecise Young's Modulus	280	62.1	337.7	592.1	118.0	239.8	100.5
	260	125.2	704.1	1230.0	221.1	475.8	202.9
	240	187.3	1094.3	1951.2	320.7	795.2	315.4
Imprecise markers position	0.05	47.7	300.2	950.6	125.9	349.1	158.3
	0.10	94.6	601.0	2088.0	245.7	712.0	314.5
	0.15	142.1	900.5	3379.2	368.8	1081.1	474.0
Imprecise actuator input	0.10	23.8	452.9	663.6	196.7	330.2	74.8
	0.15	34.5	656.9	938.6	298.2	448.8	104.6
	0.20	46.1	877.1	1267.0	396.2	613.6	138.1

solving the inverse FEM problem. The goal of introducing these errors is to evaluate the sensitivity of our approach. The performances of our strategy are shown in Tab. 4.2 and 4.3. If the model error is small, the predefined forces applied on the wrong locations are computed around zero. Therefore, only the locations where the predefined forces are computed to be a larger value are considered as the locations of external forces. If the inverse kinematic model is not accurate, the forces on the wrong locations becomes larger so that the location estimation and the force computation will be imprecise.

The 1-norm error bound of computed forces can be estimated by I_w. This can be validated by the case of imprecise markers position in Tab. 4.2 and 4.3. In addition, I_w can be employed to choose a better distribution of markers. We can validate it from the case of imprecise markers position using direct FEM $\mathbf{S}_0$ and $\mathbf{R}_0$.

Based on the comparison of data in Tab. 4.2 and 4.3, the displacement vector of markers plays a more important role for our strategy. The error generated by imprecise Young's Modulus and actuator forces can be reduced by calibration before the experiment.

4.7 Validation of Marker-based Approach by Experiments

For the implementation of our strategies, the 3D positions of several feature points should be obtained at each sampling time. We proposed an allocation algorithm based on FE model to get the positions of feature points in the first subsection. Then, the experimental setup is introduced in the second subsection. In the last two subsections, we show the performances of external force sensing algorithms.

4.7.1 Feature Point Allocation

The strategy of external force sensing is implemented using the position of feature points. At each sampling time, the feature points on the real robot should be allocated according to their corresponding ones on the FE model. Usually, this task can be realized by image identification [132] or multi-object tracking algorithm [133].

We propose a robust strategy for the allocation of feature points even in the situation of feature points loss. During the process of external force sensing, the hidden feature points have to be detected. The idea is inspired by the registration strategies for soft object in [134] where deformable registration is realized using closest point correspondences and FE model.

We define a threshold displacement for the detection. For each effector on the FE model, its closest correspondence on the real robot can be found. If the displacement between them is larger than the threshold, the corresponding point on the real robot is hidden. Then the compliance matrix and the effector vector are regenerated for the external force sensing.

More feature points should be defined than needed to sense the external forces. This allow to continue the force sensing when some points are not available. Besides, we can employ the effectors which have larger displacements for the external force sensing to increase the accuracy (see the experiment results).

4.7.2 Experimental Setup

The experimental setup for the external force sensing using the soft sheet and the soft parallel robot is shown in Fig. 4.5. Three components are needed for the implementation of the experiments: a soft object with several predefined feature points, a real-time simulated model, and a position tracking system. Both the soft sheet and the soft parallel robot are made of silicone and the Young's modulus are 290 kPa and 220kPa respectively (Both values are estimated based on the silicon materials.). Four corners of the soft sheet are fixed and the soft parallel robot is actuated by cables (In the experiment, two cable actuators are considered). Several marker points are fixed on the surface of the soft objects and detected by the commercial optical tracking system *(OptiTrack* by NatrualPoint company) with sampling frequency 100Hz and a precision of 0.1mm. According to the position of marker points, the position of the effectors on the FE model are defined. The applied forces on the real soft objects are generated by rigid weights hung on the actuated points.

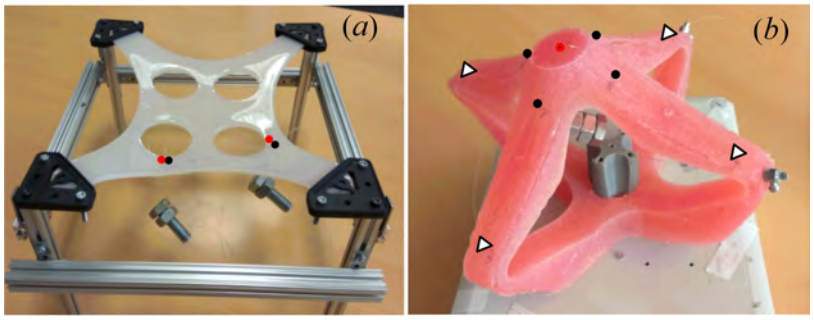

Figure 4.5: Experimental setup of soft sheet (a) and soft parallel robot (b). The black and the red points stand for the position of marker points and location of external forces respectively. Two group of markers are defined for soft robot. The triangle signs represent the group of marker points which are far from the location of external force. Several weights are employed to generate applied external forces.

4.7.3 Experiment Results of External Force Computation

For the external force computation, we assume that the locations are known. The distribution of external forces and marker points is shown in Fig. 4.5. In this subsection, we show the performances of external force computation for soft sheet and soft parallel robot respectively.

4.7.3.1 Force Computation of Soft Sheet

The performances of external force computation is shown in Tab. 4.4. Two external forces are applied on the soft sheet with known locations and directions. We employ three weights (i.e. $559mN$, $459mN$, and $359mN$) to generate different intensities of external forces. Six groups of applied forces are tested and the average error for each test is computed by $\frac{1}{2}\sum_{i=1}^{2} \mid F_i^c - F_i \mid /F_i$. The accuracy of our method to compute external forces can reach about 97%.

Table 4.4: Results of two forces computation for soft sheet. The unit of the forces is mN.

Applied forces	F_1	559	559	559	459	459	359
	F_2	559	459	359	459	359	359
Computed forces	F_1^c	577	572	570	470	471	375
	F_2^c	532	443	344	440	354	356
Average error		0.04	0.03	0.03	0.03	0.02	0.03

4.7.3.2 Force Computation of Soft Parallel Robot

As shown in Tab. 4.5, we divide the markers into two groups (four markers in each group). The external force is applied on the tip of the robot. In the experiments, we test two cases

(using different group of markers) for external force computation. In Tab. 4.5, the computed forces C_1 and C_2 use the first (close to the external force location) and the second (far from the external force location) group of marker points, respectively. Considering the influence of cable displacement, we test our algorithm with fixed cable displacement (8 *mm* for each cable) and with fixed applied force (659 *mN*). The accuracy of computed forces C_1 and C_2 can reach to be 98% and 93%, respectively.

Table 4.5: Results of one force computation for soft parallel robot. The unite of the forces is *mN*. The unit of displacement is *mm*.

Cases	Fixed cable displacement				Fixed applied force			
	Applied forces				Cable displacement			
	859	659	459	259	6	8	10	12
C_1	882	670	463	260	661	670	643	642
Error	0.03	0.02	0.01	0	0	0.02	0.02	0.03
C_2	829	621	421	230	627	621	618	613
Error	0.03	0.06	0.08	0.11	0.05	0.06	0.06	0.07

4.7.4 Experiment Results of Location Estimation

The screenshot of the location estimation is shown in Fig. 4.6. Before the experiments, candidate locations of external forces should be predefined. In our experiments, 14 markers and 12 candidate location are defined for the soft sheet. For the soft robot, we employ 9 markers and 8 candidate locations. After deleting the smaller element in the computed external force vector, the location and the intensity of external forces are estimated and shown by red arrows in Fig. 4.6. Due to the modeling error of FE model, the number of candidate locations should not be very large and the candidate forces should not generate slight displacement of markers. Using the algorithm proposed in this paper, the locations of external forces can be obtained even if the number of actual external forces is unknown. However we need to have a set of candidate locations and a sufficient number of feature points.

We also test the performance of external force computation when some markers are not available. If some markers are not available, the feature point allocation algorithm will detect them and generates a new displacement vector and compliance matrix. Several candidate locations should be predefined to employ this algorithm. The method works if the external forces are applied on the predefined locations. However, for real applications, the forces may be applied everywhere which deteriorates the performances of the algorithm. We test actual cases where the external forces are not applied on the candidate locations precisely. As shown in Fig. 4.6, if the real locations are close to some candidate ones, the algorithm will find

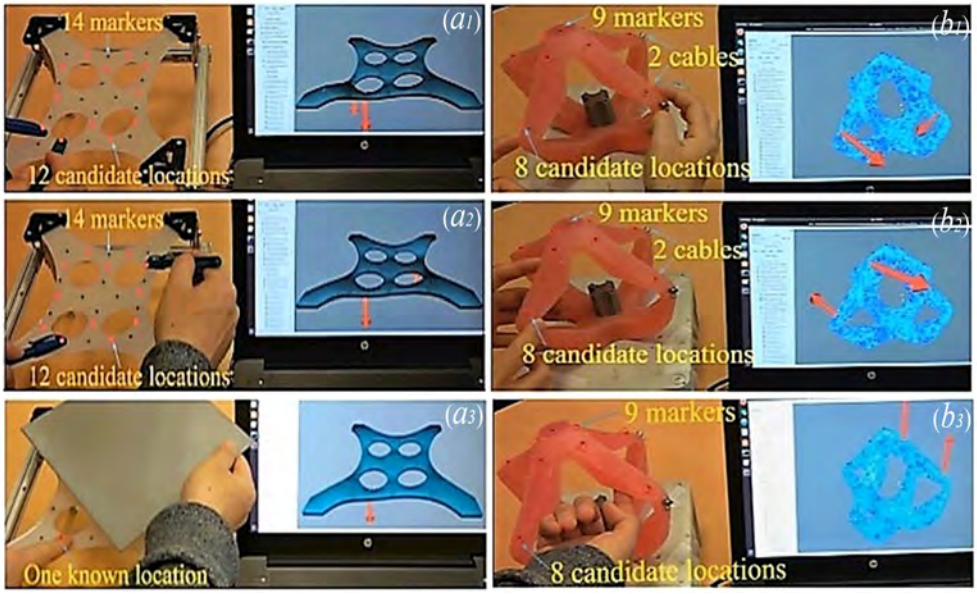

Figure 4.6: Location estimation of external force for soft sheet (*left one*: a_1, a_2 estimation of forces; a_3 force computation when some markers are hindered) and soft parallel robot (*right one:* b_1, b_2 estimation of forces; b_3 rough estimation of location).

these candidate locations as the approximation of the real ones. However, if the real locations are not very close to candidate ones, several candidate locations around the real ones will be founded so that we can get a rough estimation using the geometrical center of founded candidate locations.

4.8 Experiment Results of Marker-free External Force Sensing

The marker-free strategy is validated by some experiments described in this section. In the first subsection, the experimental setup is introduced. Then, the performances of marker-free force sensing method are shown in the second subsection.

4.8.1 Experimental Setup

As shown in Fig. 4.7, the experimental setup consists of three components: a soft parallel robot (the same robot shown in Fig. 4.5), a real-time FE model, and an RGB-D camera. The soft robot has four legs and is actuated by cables on each leg so that it is difficult to get an analytical kinematic model. It is made of silicone with the Young modulus 180kPa (The value

is estimated by simple calibration based on the relationship of the weight on the tip and the deformation.). The FE model of the soft robot is built with 4884 elements and is simulated in the SOFA framework [42], which is a real-time FEM simulation software. The methods for FEM simulation and soft robot modeling are described in details in [127]. The configuration of the soft robot is updated with the frequency of 22 Hz (5 feature points) for the calibration and 7Hz (51 feature points) for the external force sensing. The *Realsense* depth camera D415 by Intel company is employed to capture the scene. The frequency of the image processing is 10Hz for force sensing.

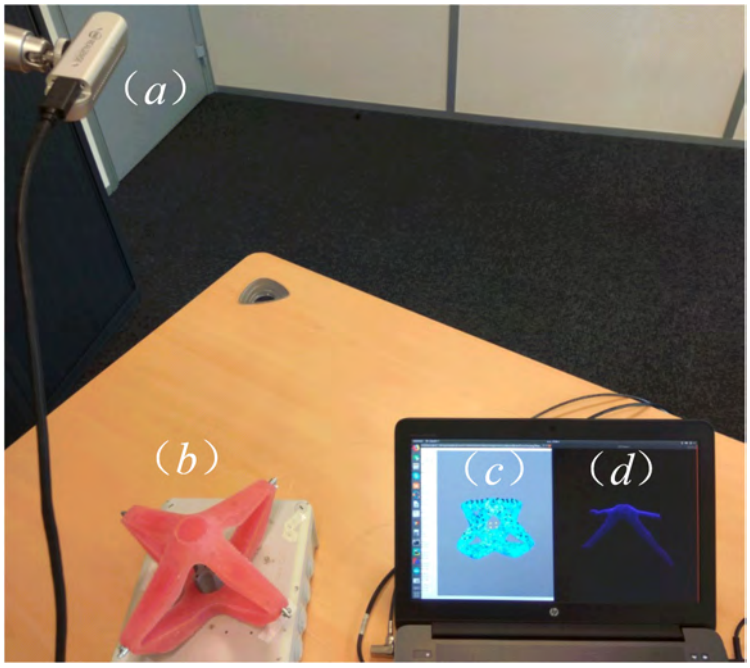

Figure 4.7: Experimental setup for calibration and external force sensing of soft robot. The setup consists of an RGB-D camera (a) which generates the point cloud of robot surface (d), a soft parallel robot (b), and a real-time FE simulator (c).

4.8.2 External Force Sensing Performances

Given the accurate initial configuration of the FE model provided by the calibration step, the external force sensing can be handled. In this subsection, we show the performances of image processing, external force computation and the compound of location detection and intensity computation.

4.8.2.1 Performance of Image Processing

The image processing step in this paper is based on some algorithms provided in the Point Cloud Library (PCL) [131]. The objective is to segment the point cloud of the robot and the one of its surrounding environment. The color-based region growing segmentation technique is employed, with the assumption that the color of the robot is homogeneous and is different with respect to its surrounding. As shown in Fig. 4.8, the contact points can be detected successfully with the blue points being the surface of the robot and the yellow points being the contacts location.

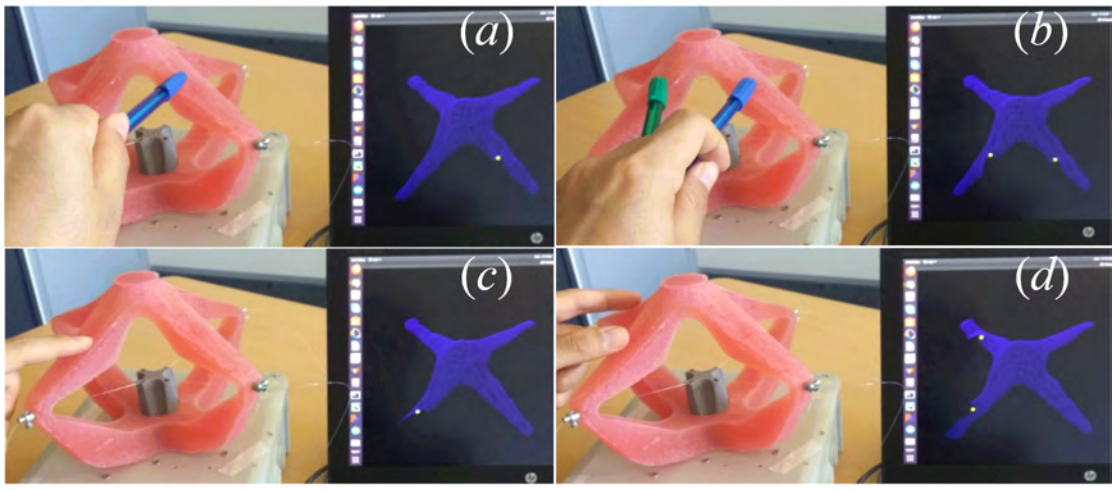

Figure 4.8: Screenshot of image processing results about the location of external forces. The yellow points are the detected location of external forces. (a) and (c) are the situation of one external forces. (b) and (d) are the detection of two external forces.

4.8.2.2 Performance of External Force Computation

Multiple three-dimensional external forces can be computed and tracked using the proposed methods. But in this subsection, we focus on one-dimensional force computation. As shown in Fig. 4.9, the external force generated by a weight in z-direction which is along the direction of gravity is applied on the tip of the robot. The control inputs are also considered with constant cable displacements. Based on the inverse FE model, the intensity of the weight is computed and is listed in Tab. 4.6 for different weight. The Young's Modulus is set to be 180 kPa which is calibrated based on the relation between weight and deformation. The initial calibration without any external forces is needed to reset the computed force to zero. From Tab. 4.6, the computed intensity can reach a higher level of accuracy where the average error for six weights is around 4%.

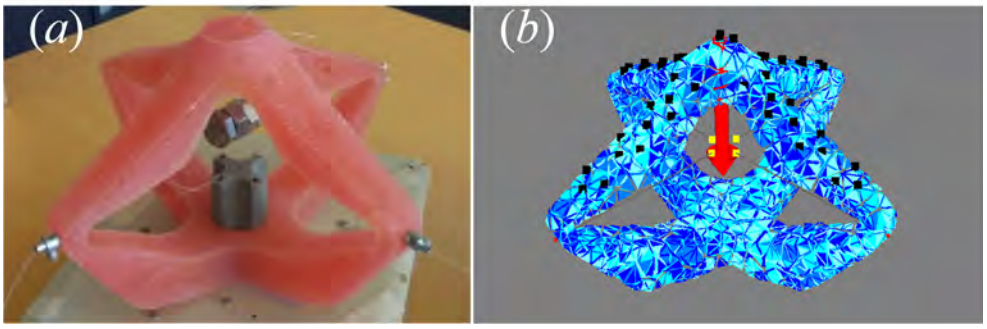

Figure 4.9: Experiment setup for the validation of intensity computation. (a) the soft robot with a weight hanging on the tip; (b) the FE model with feature points and computed force.

Table 4.6: Intensity computation of external force.

Actual force (mN)	260	460	660	760	860	1060
Computed forces (mN)	230	434	634	756	873	1045

4.8.2.3 External Force Sensing

In Fig. 4.10, the screenshots of the external force sensing system is shown with the integration of location estimation and intensity computation. In the experiment, we consider two cases: the soft robot with one external force ((a) and (c)) and two external forces ((b) and (d)). 51 feature points are pre-defined on the surface of FE model and are regarded as the possible locations of external forces. When the locations of the external forces are detected based on the image processing of the point cloud, the correct locations on the FE model are selected. Then the intensities are computed by solving the optimization problem (4.16) so that the FE model can be actuated to align to the corresponding point cloud.

4.9 Conclusion and Discussion

This chapter presents two vision-based methods (marker-based and marker-free) of external force sensing for soft robots or soft objects. Both methods allow to estimate both the locations and intensities of external forces without using embedded force sensors. For the marker-based approach, the estimation of locations and intensities is achieved in one step by solving an optimization problem. The correct locations of external forces are found among several possible ones. Without using feature points on the robot surface, the marker-free vision-based force sensing strategy is proposed with two independent steps: location detection of external forces and intensity computation.

For the vision-based external force sensing, the performances deteriorate if the material pa-

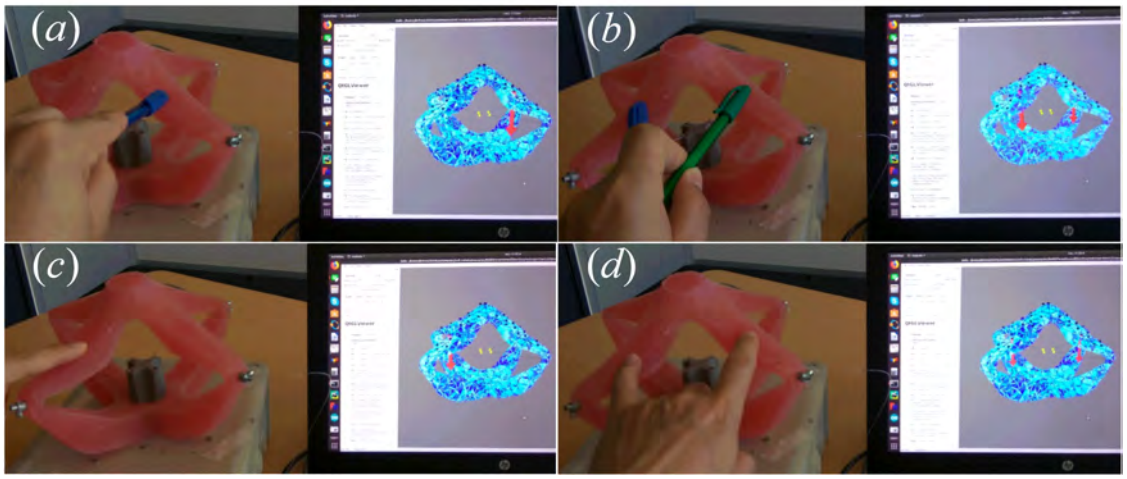

Figure 4.10: Screenshot of external force sensing. The robot has four cables with constant length for the experiment. (a) and (c) show one external force on the actuated soft robot. (b) and (d) show case with two external forces.

rameters or the position vector are not accurate. The size and the complexity of the mesh plays an important role: while we would need a very fine mesh for obtaining the convergence of the FEM method, we are constrained by the computation time if we want a real-time measurement. Due to the computation time, our method is probably not a good solution for embedded systems. Moreover, the model employed in our paper is quasi-static and the assumption is large displacements but small strain (Hooke's law). The method has not been tested with large stress using a non-linear material model. The method can not capture dynamic/viscoelastic behaviors.

Using the marker-based force sensing strategy, the placement of these features points should be carefully distributed over the surface and a sufficient number of feature points should be used to capture multiple forces, or multiple locations. For the marker-free external force sensing using an RGB-D camera, a limited number of feature points should be defined for the FE model to balance the computation time and external force sensing accuracy. We could implement the method on GPU to have better performances. We employ a color-based image segmentation technique for the point cloud which reduces the generalization and is sensitive to the color contrast between the considered areas. The external forces are assumed to be point-wise forces which are applied on the surface of the soft robot. Therefore, the estimation accuracy reduces if the contact area between the robot and it environment is larger.

Chapter 5

Position Control of Cable-Driven Continuum Catheter Robots Through Contacts

5.1 Overview

The loss of contact information for the closed-loop control design of cable-driven continuum robots could result in a positive feedback loop [44]. To deal with this problem, this chapter presents a method that aims at providing a safer insertion for cable-driven continuum catheter robots by decoupling the bending and insertion movement. The proposed strategy allows the surgeon to remain in direct control of the catheter insertion and provides the visualization of contact forces along the entire catheter.

To achieve the closed-loop control, the catheter robot and the contacts are simulated in a real-time simulator and are combined with the real system. The simulated system provides the Jacobian matrix and detects the collision between the robot and its environment. Then, both control inputs and contact forces are computed by solving a quadratic programming (QP) with a linear complementarity problem (LCP). Usually solving the problem of QP with LCP is difficult, so it is necessary to explore a simpler strategy for the optimization.

The work in this chapter is documented in [115] and this chapter is organized as follows. Section 2 introduces the quasi-static equation with contact modeling based on real-time Finite Element Method. The overview of the proposed strategy is shown in Section 3. The closed-loop decoupled control is presented in Section 4. The simulation-based validation and experiments are presented in Section 5. Finally, Section 6 summarizes the chapter.

5.2 Quasi-static Equation of Catheter Robots based on FEM

The catheter robot is modeled using beam elements [135] with the assumption that the geometry of the cross section is constant and the dimension along the length is much larger than the other dimension.

Then, the quasi-static equation of catheter robot can be deduced as the equations (2.4), (2.5) and (2.6) introduced in Chapter 2. We assume that δ_e, δ_a and δ_c are the position of effectors (the tip of the catheter robot), the displacement of actuators (the nodes where the external controlled inputs act on) and the penetration gap of contacts (the nodes where the contacts act on). The equations (2.4), (2.5) and (2.6) can be rewritten as:

$$\delta_e = \mathbf{W}_e \lambda + \delta_e^{free} \tag{5.1}$$

$$\delta_a = \mathbf{W}_a \lambda + \delta_a^{free} \tag{5.2}$$

$$\delta_c = \mathbf{W}_c \lambda + \delta_c^{free} \tag{5.3}$$

where $\mathbf{W}_i = \begin{bmatrix} \mathbf{W}_{ia}(\mathbf{x}) & \mathbf{W}_{ic}(\mathbf{x}) \end{bmatrix}$ with i standing for the subscripts e (effector), a (actuator) and c (contact), respectively. The external forces applied on the FEM can be written as $\lambda = \begin{bmatrix} \lambda_a & \lambda_c \end{bmatrix}^T$.

Based on the Signorini's law, the complementarity relation can be written as [62]:

$$\mathbf{0} \leq \delta_c \perp \lambda_c \geq \mathbf{0} \tag{5.4}$$

Compliance matrices $\mathbf{W}_e$, $\mathbf{W}_a$ and $\mathbf{W}_c$ depend on the configuration of catheter robot so that they change at each iteration and cannot be precomputed. We assume that the number of effectors, actuators and contacts are respectively n_e, n_a and n_c. If the actuators and contacts act on different FEM nodes or with different directions, the matrix $\mathbf{W}_e$ has full row rank ($n_e \leq n_a + n_c$) or full column rank ($n_e \geq n_a + n_c$), the matrices $\mathbf{W}_a$ and $\mathbf{W}_c$ have full row rank ($n_a \leq n_a + n_c$ and $n_c \leq n_a + n_c$).

5.3 Problem Statement and Overview of Proposed Strategy

In this section, we first introduce the problems for the control design without considering the contacts. Then, we give an overview of the decoupled motion control strategy for a cable-driven catheter robot through contacts.

5.3.1 Problems for Control Design without Considering Contact

As shown in Fig. 5.1 which is first introduced in [44], there are two canonical problems (artificial singularity and inverted mapping) for cable-driven continuum robots if contacts exist on the robot but are not considered for the closed-loop control design. In the worst case, unsensed contacts would result in a positive-feedback loop. Both the artificial singularity and the inverted mapping are generated by the coupling between the insertion and bending. Therefore, to avoid the two canonical problems (shown in Fig. 5.1), we propose a decoupled control strategy by decoupling the insertion and the bending. Besides, it aims to increase the insertion safety because it enables automatic bending control and manual insertion control.

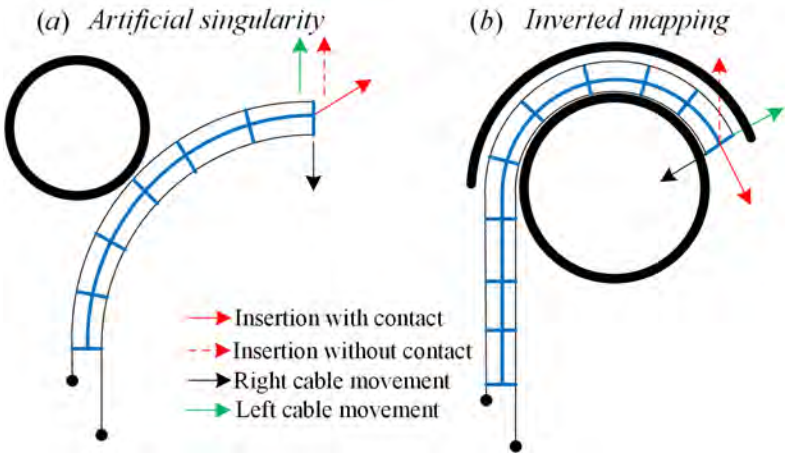

Figure 5.1: Two canonical cases described in [44] where the contacts highly influence the kinematics of the catheter robot. Arrows are the motion directions of the tip for imposed motion at the base: insertion, right cable movement, left cable movement. (a) artificial singularity and (b) inverted mapping result in the deterioration and even the positive-feedback loop.

5.3.2 Decoupled Closed-loop Bending Control

As shown in Fig. 5.2, the proposed closed-loop strategy needs to combine both the catheter robot and its FE model which is simulated in real-time. The FE model is employed to compute the Jacobian matrix and to detect collisions between the robot and the environment. The same insertion input is sent to both the catheter robot and its FE model.

For the implementation, we suppose that the physician defines a trajectory for the tip of the catheter. The closed-loop control aims at keeping the tip on this trajectory, while the physician manually inserts the catheter.

A closed-loop bending control strategy is implemented based on the quasi-static model in 2D space converted by the mapping module. Both the catheter robot and its FE model

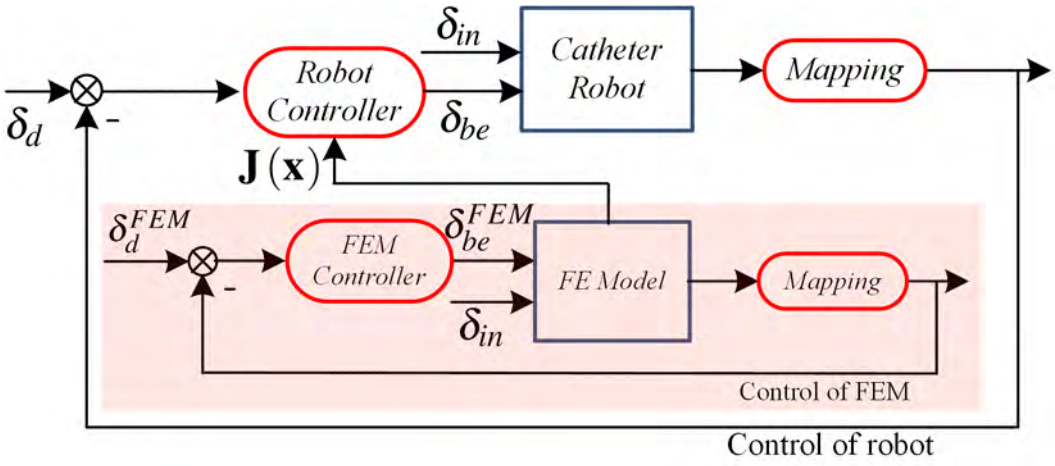

Figure 5.2: Framework of decoupled closed-loop bending control for catheter robot through contacts. For the implementation, the FE model of catheter robot and its environment is simulated in real-time. The insertion and bending are decoupled by the mapping block. Two independent controllers (Robot Controller and FEM Controller) are employed for the control of catheter robot and its FE model, respectively. δ_d and δ_{be} are respectively the desired position and bending control inputs for the catheter robot, while δ_d^{FEM} and δ_{be}^{FEM} are the corresponding variables for the FE model. δ_{in} is the insertion input for both the catheter robot and the FE model.

are controlled to follow the same predefined trajectory (δ_d and δ_d^{FEM} respectively). For the actuation of the FE model, the control inputs and the contact forces are computed by solving a minimization problem that contains a QP (for the control inputs) and a LCP (for the contact forces). Using the decoupled strategy, the robot controller is designed without taking contacts into consideration, which aims to simplify the control design and to avoid the impacts of imprecise contact model.

The advantages of this strategy are: (1) The decoupled implementation of insertion and bending provides a possible solution to increase the safety for catheter insertion and to allows the surgeon to remain in direct control of the catheter insertion; (2) The closed-loop bending control strategy provides a more accurate tracking performances; (3) It also allows to design a stable controller for continuum robot without considering the contact; (4) Contact forces along the entire catheter can be computed and visualized for the surgeons.

5.4 Decoupled Control

In this section, we introduce the design of the three components (Mapping, FEM Controller and Robot Controller) needed for the decoupled control of catheter robots.

5.4.1 Mapping Component Design

The mapping component is used to decouple the insertion and the bending for the catheter robot. As shown in Fig. 5.3, the movement of catheter robot is modeled in 3D space defined by the coordinate system $OXYZ$. To achieve the mapping, the local coordinate system $Pxyz$ is computed at each step based on the 3D tip position and the predefined trajectory so that it moves with the insertion of catheter robot. After the mapping process, the movement of catheter tip in 3D space is converted to the insertion along the tangent axis x and the bending movement in 2D plane defined by the normal axis z and the projection axis y.

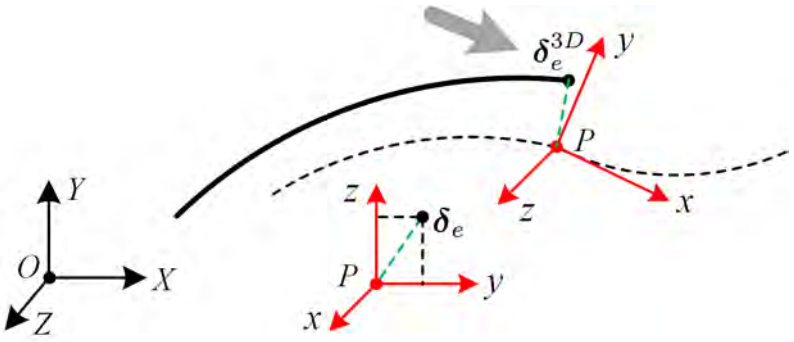

Figure 5.3: Mapping strategy for the catheter robot. The 3D position of the catheter tip in the global coordinate system $OXYZ$ is mapped to be the 2D position in the local system $Pxyz$. P is the nearest point on the trajectory to the catheter tip δ_e^{3D}. Px is the tangent vector for the trajectory at the point P. Py and Pz are respectively the projection vector and the normal vector.

The algorithm for the mapping component design is summarized in Algorithm 5.1. The 3D position δ_e^{3D} with respect to the coordinate system $OXYZ$ is computed based on the FE model. On the predefined trajectory, the nearest point P to the catheter tip δ_e^{3D} can be computed. Then, at the point P, we can compute the tangent vector Px with respect to the trajectory. The normal vector Pz is computed to be perpendicular to both Px and $\delta_e^{3D}P$. After the computation of Px and Pz , we can compute Py which is perpendicular to both Px and Pz.

We map the 3D equations (5.1), (5.2) and (5.3) on a moving 2D plane Pyz. This mapping provides modeling equations for controlling the bending movement[1]. During the bending control design, the desired position of catheter tip is the nearest point P on the predefined trajectory. Then the 3D position error $\delta_e^{3D}P$ is mapped to a 2D vector in the plan Pyz and it is norm is minimized with the bending controller.

For the implementation of the feedback bending control, the FEM controller and the Robot

[1]To simplify the notion, the equations (5.1), (5.2) and (5.3) in the following of this paper are used to capture the motion in 2D space.

Algorithm 5.1 Mapping algorithm

Initialization: Define the trajectory for the catheter tip;
For each step **do**
1. Local coordinate system computation
   Compute the 3D position δ_e^{3D} in $OXYZ$
   Compute point P and vector Px
   Compute vector Pz and vector Py
2. Map the 3D position error $\delta_e^{3D}P$ to the plan Pyz
3. Map the quasi-static equations (5.1), (5.2) and (5.3) to the plan Pyz

end

controller are introduced in the following subsections.

5.4.2 FEM Controller Design

The object of this section is to design a closed-loop controller for the FE model so that the tip of the FE model δ_e converges to its desired value δ_d[2]. We employ the optimization-based approach to compute the control inputs. The optimization problem can be defined as $\lambda = \arg\min \Gamma(\lambda)$ where $\Gamma(\lambda)$ is the objective function:

$$\Gamma(\lambda) = \frac{1}{2} \parallel \mathbf{W}_e\lambda + \delta_e^{free} - \delta_d \parallel^2 \tag{5.5}$$

For the safe interaction with the environment, the contact force is constrained as $0 \leq \lambda_c \leq \lambda_{c,max}$. Besides, the constraints of the control inputs should be also considered: $\lambda_{a,min} \leq \lambda_a \leq \lambda_{a,max}$ where $\lambda_{a,min}$ and $\lambda_{a,max}$ are chosen based on the physical limits of the actuators [3].

In order to solve the constrained optimization, the numerical optimization algorithm is necessary to be employed. In order to match the standard quadratic programing (QP) formulation for the usage of software packages [130], (5.5) can be converted to

$$\min_{\lambda} \frac{1}{2}\lambda^T\mathbf{Q}_1\lambda + \mathbf{c}_1^T\lambda \tag{5.6}$$

$$\text{s.t.} \quad \mathbf{0} \leq \lambda \leq \lambda_{max} \tag{5.7}$$

$$\text{and} \quad \mathbf{0} \leq \delta_c \perp \lambda_c \geq \mathbf{0} \tag{5.8}$$

where $\lambda_{max} = \begin{bmatrix} \lambda_{a,max} & \lambda_{c,max} \end{bmatrix}^T$, $\mathbf{Q}_1 = \mathbf{W}_e^T\mathbf{W}_e$ and $\mathbf{c}_1 = \mathbf{W}_e^T\left(\delta_e^{free} - \delta_d\right)$. For the opti-

[2]The variables in this subsection represent the variables for the FE model. However, we use the same variables for the real robot in the next subsection.

[3]For the cable-driven robots, $\lambda_{a,min} = 0$

mization, both the control inputs and the contact forces are treated as optimization variables. The linear complementarity problem (LCP) (5.8) is employed to capture the contact constraint.

In the following, we show a simplified method to solve the QP problem with LCP (QPCC).

The complementarity relation (5.4) can be written as $\delta_c^T \lambda_c = 0$ where $\delta_c \geq \mathbf{0}$ and $\lambda_c \geq \mathbf{0}$. Finding a solution to the LCP (5.8) is associated with minimizing the quadratic function

$$f(\lambda_c) = \lambda^T \mathbf{W}_c^T \lambda_c + \left(\delta_c^{free}\right)^T \lambda_c \tag{5.9}$$

subject to the constraints $\mathbf{W}_c \lambda + \delta_c^{free} \geq 0$ and $\lambda_c \geq 0$.

Considering that $\lambda_c = \begin{bmatrix} \mathbf{0}_{c\times a} & \mathbf{I}_{c\times c} \end{bmatrix} \lambda$, the quadratic function (5.9) can be rewritten as

$$f(\lambda) = \lambda^T \mathbf{Q}_{lcc} \lambda + \mathbf{c}_{lcc}^T \lambda \tag{5.10}$$

where $\mathbf{Q}_{lcc} = \begin{bmatrix} 0_{(a+c)\times a} & \mathbf{W}_c^T \end{bmatrix}$ and $\mathbf{c}_{lcc} = \begin{bmatrix} 0_{1\times a} & \delta_c^{free} \end{bmatrix}^T$.

Then the QP problem with a LCP (5.6) can be separated into two standard QP problems:

$$\begin{aligned} \text{QP1}: \quad & \min_{\lambda} \frac{1}{2} \lambda^T \mathbf{Q}_1 \lambda + \mathbf{c}_1^T \lambda \\ & \text{s.t.} \quad \mathbf{0} \leq \lambda \leq \lambda_{max} \end{aligned} \tag{5.11}$$

and

$$\begin{aligned} \text{QP2}: \quad & \min_{\lambda} \frac{1}{2} \lambda^T \mathbf{Q}_2 \lambda + \mathbf{c}_2^T \lambda \\ & \text{s.t.} \quad \mathbf{W}_c \lambda + \delta_c^{free} \geq 0 \\ & \text{and} \quad \lambda \geq 0 \end{aligned} \tag{5.12}$$

where $\mathbf{Q}_2 = \mathbf{Q}_{lcc} + \mathbf{Q}_{lcc}^T$ (Because $\lambda^T \mathbf{Q}_{lcc} \lambda$ is a scalar, we have $\left(\lambda^T \mathbf{Q}_{lcc} \lambda\right)^T = \lambda^T \mathbf{Q}_{lcc}^T \lambda = \lambda^T \mathbf{Q}_{lcc} \lambda$.) and $\mathbf{c}_2 = \mathbf{c}_{lcc}$. The QP problem (5.12) corresponds to the LCP.

Combining both (5.6) and (5.12), the QP problem with the LCP can be converted into a standard QP problem without the LCP.

$$\text{converted QP}: \quad \min_{\lambda} \frac{1}{2}\lambda^T \mathbf{Q}\lambda + \mathbf{c}^T\lambda \tag{5.13}$$

$$\text{s.t.} \quad \mathbf{0} \leq \lambda \leq \lambda_{max} \tag{5.14}$$

$$\text{and} \quad \mathbf{W}_c\lambda + \delta_c^{free} \geq \mathbf{0} \tag{5.15}$$

where $\mathbf{Q} = \mathbf{Q}_1 + \varepsilon\mathbf{Q}_2$ and $\mathbf{c} = \mathbf{c}_1 + \varepsilon\mathbf{c}_2$ with ε being the weight of the LCP.

For general problems, the parameter ε should be chosen as a large value to make sure that the LCP holds during the optimization. Unlike former solvers [136, 137], the proposed strategy has easier background, faster computation and can employ standard QP solvers to solve the problem of QP with the LCP. However, we make the assumption that the defined trajectory is always reachable. In practice, with assumption of static environment, we plan it off-line and make sure that it is always reachable by our robot. For the case with a reachable trajectory, the solution of (5.13) is equal to the solution of (5.6) if $\varepsilon > 0$ (The proof is presented as the appendix in this end of this chapter).

5.4.3 Robot Controller Design

The quasi-static equations (5.1) and (5.2) can be converted as the discrete-time equations (refer to [85] for detail):

$$\delta_{e,k+1} = \delta_{e,k} + \mathbf{W}_{ea}(\mathbf{x}_k)\triangle\lambda_{a,k} \tag{5.16}$$

$$\delta_{a,k+1} = \delta_{a,k} + \mathbf{W}_{aa}(\mathbf{x}_k)\triangle\lambda_{a,k} \tag{5.17}$$

Combining (5.16) and (5.17), we have

$$\delta_{e,k+1} = \delta_{e,k} + \mathbf{J}(\mathbf{x}_k)\triangle\delta_{a,k} \tag{5.18}$$

where $\mathbf{J}(\mathbf{x}_k) = \mathbf{W}_{ea}(\mathbf{x}_k)\mathbf{W}_{aa}^{-1}(\mathbf{x}_k)$ is the Jacobian matrix of the catheter robot.

As shown in Section 4, it is not necessary to consider the contact information for the robot controller design. Therefore, the robot controller is designed for the bending control of catheter robot based on the kinematic equation (5.18) which allows to control the position of catheter robot by changing the displacements of actuators.

By defining a new control vector $U_k = \mathbf{J}(\mathbf{x}_k)\triangle\delta_{a,k}$, the kinematic model can be rewritten

as

$$\delta_{e,k+1} = \delta_{e,k} + U_k \tag{5.19}$$

The tracking error is defined as $e_k = \delta_{e,k} - \delta_{d,k}$. In the task space, the control vector U_k can be designed as:

$$U_k = -K_P e_k - K_D (e_k - e_{k-1}) \tag{5.20}$$

where K_P and K_D are constant parameters for the proportional gain and differential gain respectively.

Then, the Pseudo-inverse based control allocation is employed to obtain a unique solution:

$$u = \mathbf{J}^+ (\mathbf{x}_k) U_k \tag{5.21}$$

which is the minimum 2-norm solution of $U_k = \mathbf{J}(\mathbf{x}_k) \triangle\delta_{a,k}$ and $u = \triangle\delta_{a,k}$ is the control input of the robot. For over-actuated system, the contribution of actuators with minimal energy can be computed by (5.21). Considering that the matrix $\mathbf{J}(\mathbf{x}_k)$ has full row rank (There are more actuators than effectors in our experiment.), the Pseudo-inverse matrix $\mathbf{J}^+(\mathbf{x}_k)$ can be computed by $\mathbf{J}^+(\mathbf{x}_k) = \mathbf{J}^T(\mathbf{x}_k)\left(\mathbf{J}(\mathbf{x}_k)\mathbf{J}^T(\mathbf{x}_k)\right)^{-1}$.

Substituting (5.20) into the (5.21), the control law can be written as:

$$u = -\mathbf{J}^+ (\mathbf{x}_k) [K_P e_k + K_D (e_k - e_{k-1})] \tag{5.22}$$

5.5 Experiment Results using Decoupled Control

In this section, we first introduce the experimental setup and the method of sensor-robot calibration for the validation of the proposed decoupled control strategy. Then, we show the experiment results of trajectory tracking and positioning for a cable-driven continuum catheter robot. Besides, we validate the collision detection and contact forces estimation by experiment using a soft rod. Finally, we test respectively the decoupled control with Jacobian matrix rotation and the trajectory tracking through a soft tube by simulation.

5.5.1 Experimental Setup

The experimental setup for the position control of a cable-driven catheter robot through contacts is shown in Fig. 5.4. Three components are needed for the implementation of the experiments: a position tracking system (a), an obstacle to generate the contacts between catheter

Figure 5.4: Experimental setup for the motion control of a catheter robot through contacts. (a) NDI electromagnetic tracking system: Aurora; (b) a rigid tube for the experiment. The white points are the feature points which are used for the calibration; (c) the cable-driven continuum catheter robot which has four step motors to actuate the bending and one step motor to control the insertion.

robot and its environment (b), and a continuum catheter robot (c).

The commercial electromagnetic tracking system is employed *(Aurora* by NDI company) with the sampling frequency 20Hz and the sub-millimetric accuracy. A rigid tube (diameter 28*mm*) is fixed with respect to the base of the catheter robot. Positions of seven feature points are captured by the tracking system and are employed for the calibration between the coordinate system of the sensor and that of the robot. The movement of catheter robot consists of bending and insertion. The bending is controlled by four cables using four step motors and the insertion is actuated using one step motor. The robot is a modified version in [138]. The structure of catheter robot (diameter 7*mm*) is made of polymer material which is MR-compatible and is manufactured using the rapid prototyping technique. Both the robot and the rigid tube are simulated in SOFA framework (a real-time FEM simulation software). The frequency of the simulation can reach to 50Hz in our experiment (100 mesh nodes). The detail strategy for FE modeling can be seen in [127].

5.5.2 Sensor-robot Calibration

When the coordinate system of the *Aurora* system and that of the catheter robot do not match, a calibration is performed. For the implementation of the calibration, we choose two corresponding sets (seven points for each set) of key points defined for the rigid tube. The positions of the keypoints are measured respectively from the *Aurora* system (A) and the FE model (B).

The correspondences between the two sets of points generates a linear least-squares problem that can be solved robustly using the Singular Value Decomposition (SVD) method [139]. Then, the transformation matrix between two sets of points can be estimated. The optimal

rigid transformation can be obtained by minimizing the following error:

$$err = \sum_{i=1}^{n} \| \mathbf{R}P_A^i + \mathbf{T} - P_B^i \|^2$$

Where **R** and **T** are the rotation matrix and translation vector applied to point set A. The application of the calibration can be used as long as there are at least three points for each set.

5.5.3 Trajectory Tracking Results

The central line of the tube is defined as the desired trajectory for the robot tip. The tracking performances are illustrated in Fig. 5.5 where both the desired trajectory and the catheter tip trajectory are mapped to the XZ plane and XY plane. In our experiments, the desired trajectory consists of twice bending along the insertion direction. Three different speeds of insertion (For each case, the insertion speed is uniform.) are tested in the experiments. We choose the same control parameters for all cases, $K_P = 0.2$ and $K_D = 0.002$.

The tracking errors are defined as the distance between the catheter tip and its desired position. We compute the tracking error based on the positions on the bending plane. As shown in Fig. 5.6, the average tracking error is around $0.3mm$ when the desired trajectory is a straight line. However, the higher insertion speed deteriorates the tracking performances when there is a bending along the desired trajectory. The largest error increases to be $2mm$, $3.2mm$ and $4.8mm$ respectively for the insertion speeds $0.6mm/s$, $1.2mm/s$ and $2.4mm/s$. The error increases with the increment of the insertion speed because the lower insertion speed allows longer time for the convergence of the tip position. For the application, the insertion is controlled manually and its speed should be lower when the catheter robot starts to bend.

5.5.4 Positioning Results

The positioning results are shown in Fig. 5.7 which includes two cases with different catheter length ($60mm$ and $166mm$). The tracking error converges faster when the length of catheter is shorter. For both cases, the positioning errors converge to a small bound (around $0.2mm$). Usually a longer length of catheter robot generates a larger convergence bound if there is no interaction with the environment. However, in our experiments, the contacts limit the deflection and therefore reduce the convergence bound.

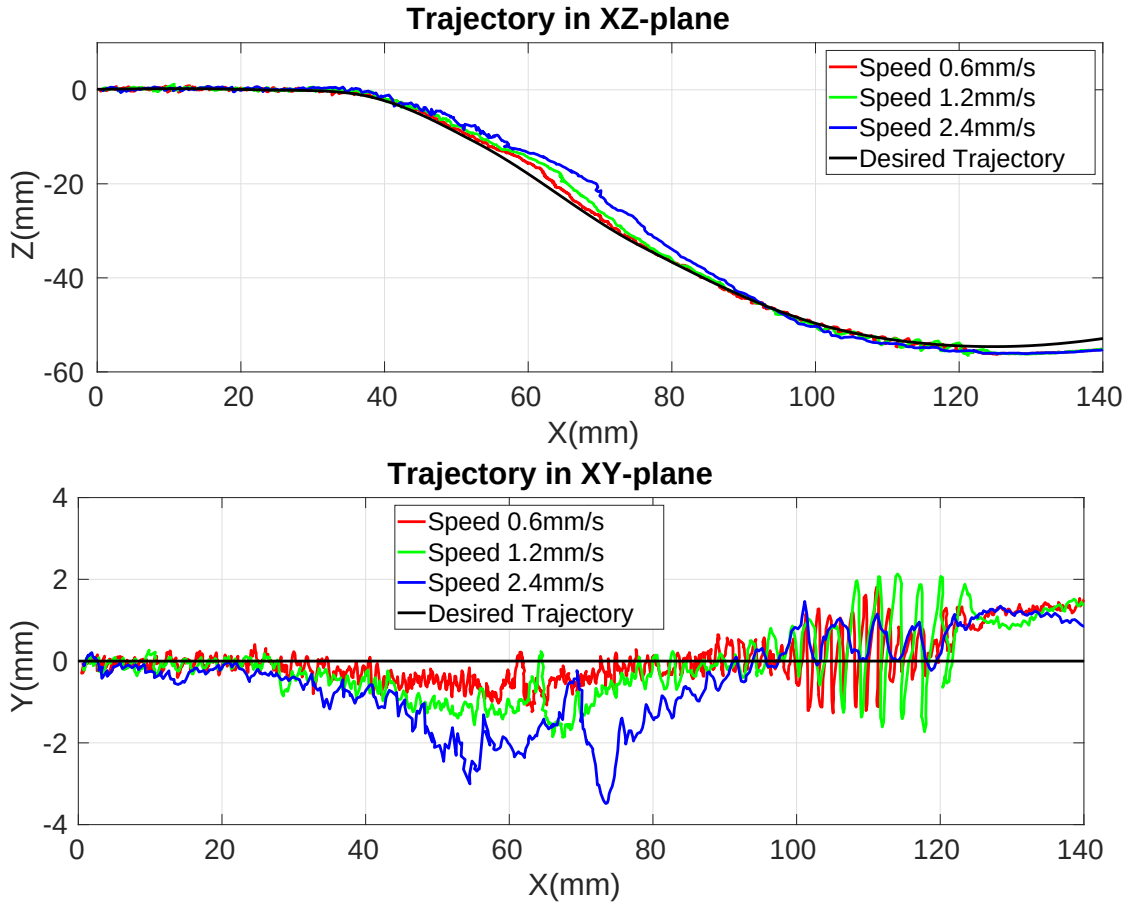

Figure 5.5: Trajectories of catheter tip in XZ plane and in XY plane with different speeds of insertion ($0.6mm/s$, $1.2mm/s$ and $2.4mm/s$).

5.5.5 Estimation Results of Contact Forces

By solving a QP problem with the LCP in (5.13), the contact forces on the entire robot can be computed. In Fig. 5.8, we choose three configurations of the interaction between the robot and its environment. The contact forces are applied on the node of FE model. There are one ($0.073N$), two ($0.079N$ and $0.122N$) and three ($0.223N$, $0.301N$ and $0.004N$) contact forces for the configurations (a), (b) and (c), respectively. Considering that the FE model is simulated to be synchronous with the catheter robot, the contact forces computed based on the FE model can be regarded as the approximation of the contact forces for the catheter robot.

5.5.6 Validation using a Soft Rod

As shown in Fig. 5.9, three experiments using a soft rod are designed to validate respectively the collision detection, the insertion in 3D tube and the contact force computation. The col-

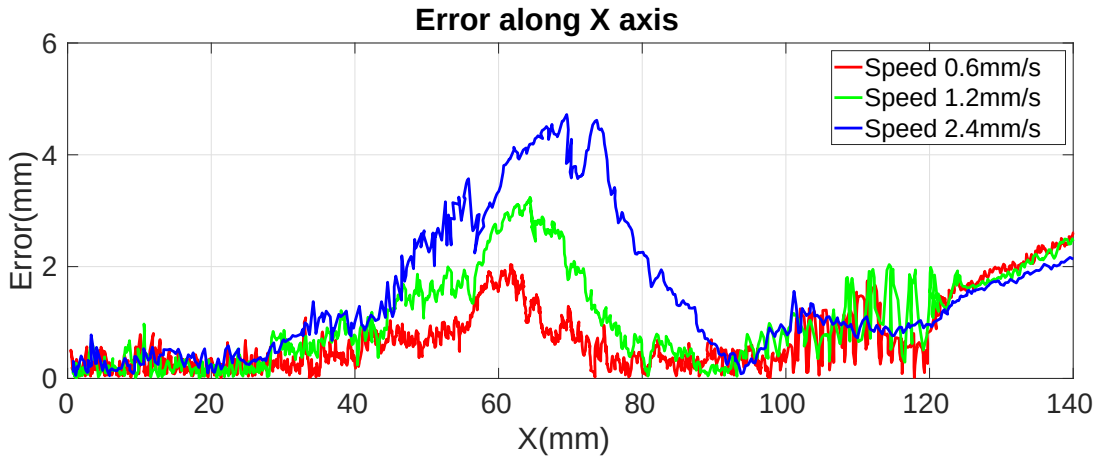

Figure 5.6: Trajectory errors of catheter tip with different speeds of insertion.

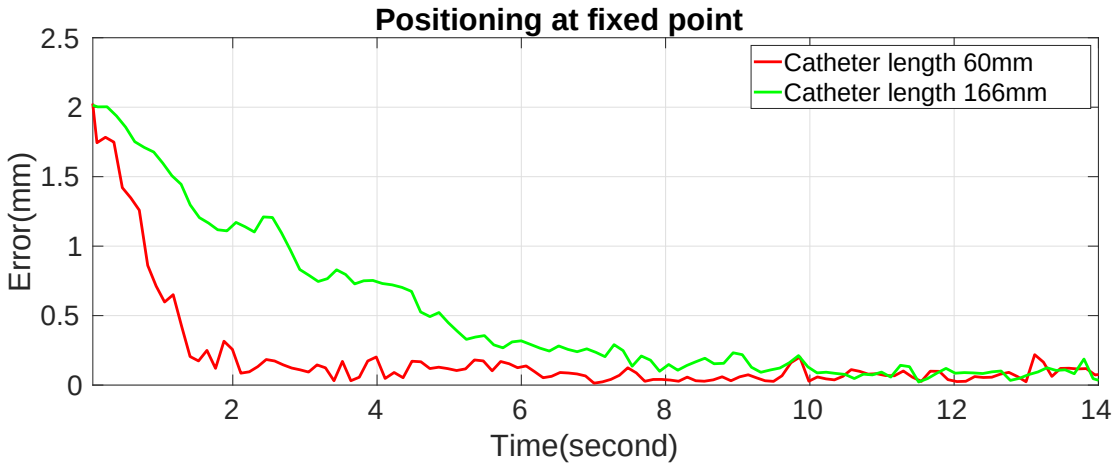

Figure 5.7: Convergence of positioning errors with different catheter length (60*mm* and 166*mm*).

lision is detected based on the relative positions between the contact and the soft rod. The contact positions are very close for both the FE model and the real system (see (a) and (b) in Fig. 5.9). The FE model is tested using a 3D tube (see (c) and (d) in Fig. 5.9). Ten static positions are measured with different insertions and compared between the real system and the FE model. The positions are drawn in Fig. 5.10 and the average position error is 3.35mm in our experiment. The accuracy of the FE model ensures the success to capture the relation between the insertion and the tip position.

The proposed method can estimate the distributed contact forces. However, the validation of this situation is very difficult. In order to simplify the experiment, we consider the case of

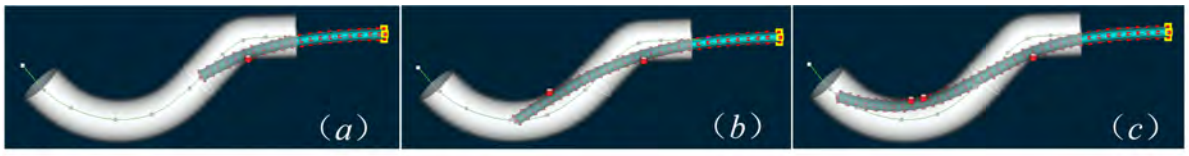

Figure 5.8: Computation of contact forces along the entire catheter robot for three configurations (a), (b) and (c).

one contact force acting on any position of the soft rod. The computation of contact force is validated using the setup (e) and (f) in Fig. 5.9. The length of the soft rod is $210mm$ and its Young's Modulus is $1.4GPa$. Nine groups of forces are measured and computed with different target tip positions and different contact points. The measured forces using a force sensor and the computed forces using the FE method are listed in Tab. 5.1. The accuracy of contact force computation can reach to 97.1%.

Table 5.1: Validation of contact force computation. The unit of the forces is mN.

Target	B_1			B_2			B_3		
Contact	A_1	A_2	A_3	A_1	A_2	A_3	A_1	A_2	A_3
Measured Force	110	198	372	216	357	682	336	537	998
Estimated Force	119	193	380	214	345	679	324	527	1030

The contact forces computed based on the FE model can be regarded as the approximation of the contact forces for the catheter robot. However, the proposed method highly depends on the accuracy of FE model. To get an accurate estimation of contact forces, we need to model both the contact object and the robot with a high accuracy. This is the limitation of the proposed method.

5.5.7 Simulation-based Validation

The simulation-based validation allow to focus on the control algorithms themselves and can get rid of the sensor noise, imprecise modeling of robot and contacts. In this subsection, we first validate the decoupled control strategy to deal with the problem of the Jacobian matrix rotation. Then, we show the simulation result when the catheter robot interacts with a soft tube.

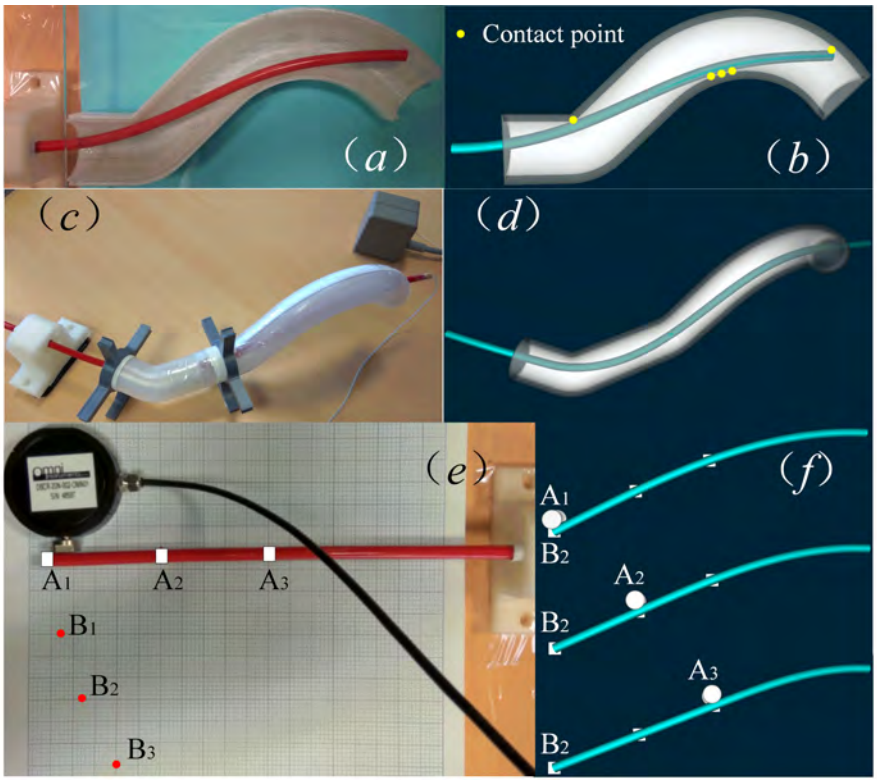

Figure 5.9: Experiment setup for the validation of the contact detection ((a) and (b)), the accuracy of simulation ((c) and (d)) and the contact force computation ((e) and (f)). In (e) and (f), A1-A3 and B1-B3 are respectively the contact locations and the desired positions of the tip.

5.5.7.1 Control Strategy to Deal with the Rotation of Jacobian Matrix

As shown in Fig. 5.11, the catheter robot can be controlled from the initial configuration *(a)* to the finial configuration *(b)* with the decoupled control strategy. If the insertion and bending are coupled between each other, the Jacobian matrix is inverted during the movement. However, the decoupled control strategy can successfully avoid the problem of the Jacobian matrix rotation when the robot interacts with the contacts.

5.5.7.2 Decoupled Control with Soft Tube

The trajectory tracking can also be achieved when the catheter robot interacts with soft contacts. The decoupled control strategy is employed and validated using simulation which is shown in Fig. 5.12. To estimate the Jacobian matrix and contact forces, both catheter robot and soft tube are modeled using FE method and are simulated in real-time.

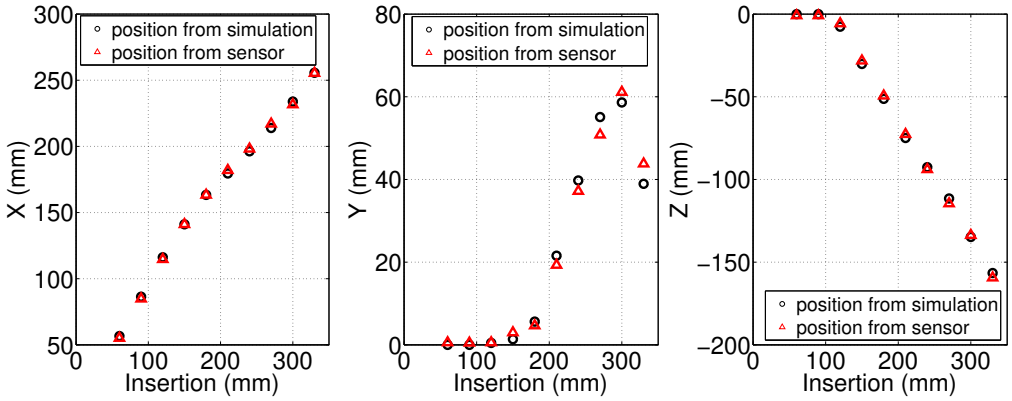

Figure 5.10: Comparison of tip positions for the experiment in Fig. 5.9 ((c) and (d)).

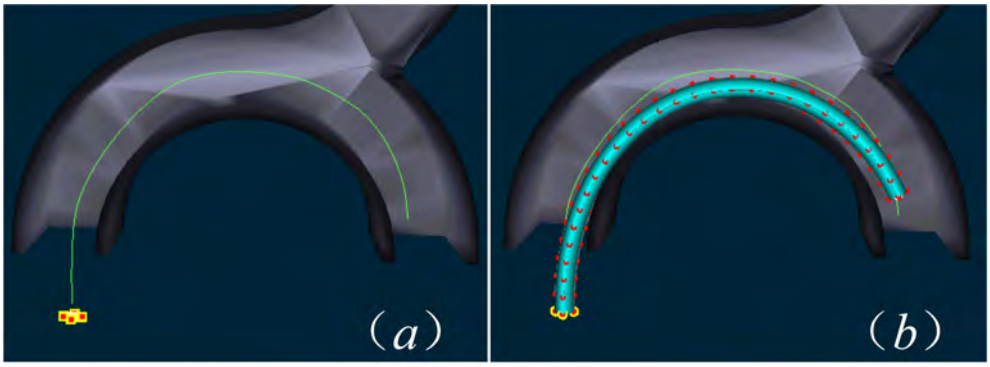

Figure 5.11: Simulation-based control validation for catheter robot through contacts. (a) and (b) are respectively the initial configuration and the finial configuration of the catheter robot.

5.6 Conclusion and Discussion

Catheter-based intervention plays an important role in minimally invasive surgery. The closed-loop bending control is explored for a cable-driven continuum catheter robot through contacts. A decoupled control strategy is proposed, which allows to control insertion and bending independently. This method can provide a stable control and the information of contact forces along the entire catheter. The catheter robot is modeled based on FE method where the contact is treated as a linear complementarity problem (LCP). Then, the motion control is converted to a quadratic programming (QP) problem with complementarity constraints (QPCC). A simplified solution is proposed for the computation of QPCC by converting it into a standard QP problem where the LCP holds strictly when the desired trajectory is reachable.

To estimate the contact forces for the catheter robots, we need the assumption that we have an accurate model of the contact and the tip position of the catheter. However, it is very difficult to always fulfill this assumption in complex situation of real applications. When the

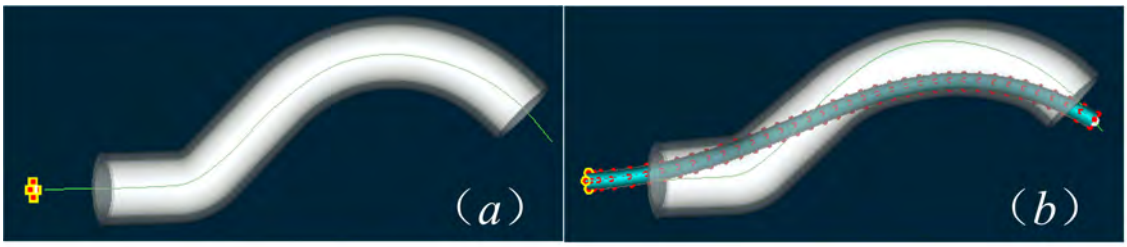

Figure 5.12: Simulation-based validation of decoupled control of catheter robot through a soft tube. (a) The initial state of catheter robot and the soft tube; (b) The simulated deformation of the soft tube and the robot.

contact objects are not modeled well, the registration method for deformation structures [79] could be employed to improve the accuracy of FE model. Besides, to make the tracking error as small as possible, the insertion speed should be limited. When the catheter robot starts to bend, the tracking error increases and needs relatively longer time to converge. Instead of limiting the insertion speed in a quite small range, we could use adaptive control parameters for the control law to increase the convergence speed.

Appendix

It is assumed that F_1, F_2, F_3, and F_4 are the minimum value of objective functions for (5.6), (5.11), (5.12) and (5.13), respectively. λ_1, λ_2 , λ_3 and λ_4 are respectively the solution of QP problems (5.6), (5.11), (5.12) and (5.13). Then, we have the following corollary:

Corollary: If the desired position is reachable (The solution of (5.5) is zero.), the solution of (5.13) is equal to the solution of (5.6).

Proof: The proof consists of two steps:

Step 1: $\lambda_4 = \lambda_1$ is the solution of the QP problem (5.13).

For the QP problem (5.6) with the solution λ_1, we have

$$F_1 = -\frac{1}{2} \parallel \delta_e^{free} - \delta_d \parallel^2 \tag{5.23}$$

and the LCP (5.8) holds so that, for QP problem (5.12) with $\lambda_3 = \lambda_1$, we have

$$F_3 = 0 \tag{5.24}$$

Then, for the QP problem (5.11) with $\lambda_2 = \lambda_1$, we have

$$F_2 = -\frac{1}{2} \parallel \delta_e^{free} - \delta_d \parallel^2 \tag{5.25}$$

Therefore, the objective functions (5.13) has the same minimum value

$$F_4 = -\frac{1}{2} \parallel \delta_e^{free} - \delta_d \parallel^2 \tag{5.26}$$

and $\lambda_4 = \lambda_1$ is the solution of the QP problem (5.13).

Step 2: $\lambda_1 = \lambda_4$ is the solution of the QP problem (5.6).

Because the objective function (5.13) is the sum of (5.11) and (5.12) and have more strict constraints, we have the conclusion that

$$F_4 \geq F_2 + F_3 \geq -\frac{1}{2} \parallel \delta_e^{free} - \delta_d \parallel^2 \tag{5.27}$$

If the desired position is reachable, there is solution λ_4 for (5.13) so that its objective function has the minimum value:

$$F_4 = -\frac{1}{2} \parallel \delta_e^{free} - \delta_d \parallel^2 \tag{5.28}$$

Then, we have $F_2 = -\frac{1}{2} \parallel \delta_e^{free} - \delta_d \parallel^2$ with $\lambda_2 = \lambda_4$, and $F_3 = 0$ with $\lambda_3 = \lambda_4$. Therefore, λ_4 is a solution to make (5.6) reach the minimum value and make (5.8) hold. Then, we can conclude that $\lambda_1 = \lambda_4$ is the solution of the QP problem (5.6).

Chapter 6

Conclusion

6.1 Summary

This thesis contributes to the modeling, calibration, position control and force sensing for soft robots based on Finite Element Model. The proposed methods are general and can be employed for a wide variety of soft robots or soft objects even with complex configurations.

Chapter 3 presents a general method for the kinematic modeling, vision-based calibration and closed-loop control of soft robots based on Finite Element Model. By choosing several corresponding feature points on the surface of a soft robot and its FE model, we can simultaneously calibrate the sensor-robot system and the initial actuator inputs. The calibration is based on the numerical model of the soft robot and can be employed for robots with complex structures. The robot is simulated in real time to predict the Jacobian matrix and to generate an open-loop controller. Two closed-loop control strategies are proposed for the position control of soft robots by combining the FE model and the soft robot. Considering the error of the FE model, the robust stability is proved based on Lyapunov stability theory. Further, to deal with the problem of image feature loss during the visual servoing, the open-loop control and closed-loop control are combined to generate a switched strategy.

Chapter 4 proposes two vision-based methods (marker-based and marker-free) to measure external forces applied on soft body and soft robots. Both algorithms can be employed for complex structure, 3D workspace, force sensing on whole soft body, and multi-force actuation. For the implementation of the marker-based method, several markers are put on the surface of the soft robot or the soft object. Then, the forces are estimated by solving an optimization problem using the detected displacement of markers. The marker-free external force sensing relies on matching the point cloud provided by an RGB-D camera and predefined feature points on the surface of the FE model. Without using markers on the robot surface, marker-free method is more practical for applications. Using marker-based or marker-free methods,

both the location and the intensity of external forces on the entire surface can be estimated.

Chapter 5 contributes to the position control of catheter robot through contacts. With the goal to provide safe insertion by catheter robot, a decoupled controller is designed and allows to control the bending and the insertion independently. The proposed strategy can achieve a stable position control and can avoid the problem of Jacobian matrix rotation. As a benefit of the proposed method, the contact forces along the entire catheter robot can be computed without using force sensors. A simplified method to compute quadratic programming problem with a linear complementarity problem is proposed in this chapter. The proposed strategy is general and the application is not limited for the cable-driven catheter robot. Experiments and simulations are performed to test the proposed method.

6.2 Future Work

We discuss the future work about the calibration, position control and force sensing of soft robots in this section:

6.2.1 Calibration

The thesis employs a simultaneous approach for the calibration of sensor-robot system and the actuators. For the implementation, we need to select the feature points on the robot surface manually.

For the future work, we would like to extend the strategy for the simultaneous calibration considering material parameters (like Young's Modulus). It is also necessary to explore more effective method for the optimization. Besides, to achieve an automatic calibration, it is necessary to improve the feature points selection process instead of the manual selection strategy.

The proposed calibration is tested only using a parallel soft robot. In the future work, we would like to use the method to calibrate different types of soft robots and test it for real applications.

6.2.2 Position control

We employ proportional control laws for the closed-loop control of soft robots. The control strategy works nice when the robots have a lower speed and the FE models have a high accuracy. To deal with the limitation of lower speed, it is possible to design controllers based on the dynamic model. We would like to actuate the FE model based on the registration of the

FE model and the real robot. Using this strategy, we could increase the accuracy of FE model especially when the robot interacts with the environment.

Instead of proportional control, more advanced control methods are needed to increase the robustness and adaptivity of the control systems. For example, the adaptive control could be useful to compensate the model errors by adjusting the control parameters or control laws. We could also employ the robust control to increase the stability of the closed-loop systems. Besides, the constraints on the actuators and the workspace will be considered in the position control and path planning.

The applications of proposed methods to different soft robots (like soft manipulators and soft mobile robots) would be explored in the future work.

6.2.3 Force sensing

The vision-based force sensing methods could be improved by increase of the computation speed and the accuracy of FE model. We would like to investigate the use of model order reduction techniques [140] to have a better trade-off between precision and computation performances. The GPU programming would be employed for the simulation and image processing to increase the response speed of the force sensing algorithm.

Another improvement could be to use more precise FE model. One direction is to use more complete constitutive laws, like hyper-elastic models to test our algorithm on situations with more strain and an other direction would be the use of elements with more nodes (like cubic and quadratic tetrahedrons).

For the marker-based approach, we would like to detect the feature points based on the image processing rather than putting several artificial markers. The improvement could be very useful for the soft robots where obvious features can be found.

For the marker-free approach, the segmentation technique employed in this thesis is based on the assumption that the color of the robot is different with its contact objects. Therefore, it is necessary to improve the segmentation of point cloud using more general method.

We believe the force sensing algorithms could be useful in applications where it is complex to place force sensors on the robot, like for medical applications. We would like to evaluate the proposed techniques via concrete real world applications.

6.2.4 Catheter robots

The application of the proposed control strategy for catheter robots needs to model both the robot and the contact objects. Even if the robot can be modeled in a high accuracy, it extremely difficult to have a accuracy model of the contact objects especially when the contract objects

are changing in dynamic environment. For the motion control of catheter robots, we would like to improve the model by registration of both contacts and catheter robot with their FE models. The improvement of the FE models could increase the control performances and the accuracy of external force sensing for catheter robots.

We also would like to explore more control laws to increase the tracking performances of catheter robots. Instead of limiting the insertion speed in a quite small range, we could use adaptive control parameters for the control law to increase the convergence speed. Combining the force sensing and motion control proposed in this thesis, it is possible to design a hybrid force and position control for the catheter robot.

It is necessary to validate the proposed strategy for real applications and extend it for more challenging applications, like cardiac ablation task.

References

[1] D. Trivedi, C. D. Rahn, W. M. Kier, and I. D. Walker, "Soft robotics: Biological inspiration, state of the art, and future research," *Applied bionics and biomechanics*, vol. 5, no. 3, pp. 99–117, 2008. 1

[2] T. Umedachi, V. Vikas, and B. Trimmer, "Softworms: the design and control of non-pneumatic, 3d-printed, deformable robots," *Bioinspiration & biomimetics*, vol. 11, no. 2, p. 025001, 2016. 1, 2

[3] A. Arienti, M. Calisti, F. Giorgio-Serchi, and C. Laschi, "Poseidrone: design of a soft-bodied rov with crawling, swimming and manipulation ability," in *Oceans-San Diego, 2013*, pp. 1–7, IEEE, 2013. 1, 2

[4] J. Fras, Y. Noh, M. Macias, H. Wurdemann, and K. Althoefer, "Bioinspired octopus robot based on novel soft fluidic actuator," in *International Conference on Robotics and Automation*, 2018. 1, 2

[5] R. K. Katzschmann, J. DelPreto, R. MacCurdy, and D. Rus, "Exploration of underwater life with an acoustically controlled soft robotic fish," 2018. 1, 2

[6] T. Li, G. Li, Y. Liang, T. Cheng, J. Dai, X. Yang, B. Liu, Z. Zeng, Z. Huang, Y. Luo, *et al.*, "Fast-moving soft electronic fish," *Science Advances*, vol. 3, no. 4, p. e1602045, 2017. 1, 2

[7] A. Ramezani, S.-J. Chung, and S. Hutchinson, "A biomimetic robotic platform to study flight specializations of bats," *Science Robotics*, vol. 2, no. 3, pp. Art–No, 2017. 1, 2

[8] D. Rus and M. T. Tolley, "Design, fabrication and control of soft robots," *Nature*, vol. 521, no. 7553, p. 467, 2015. 1

[9] J. Burgner-Kahrs, D. C. Rucker, and H. Choset, "Continuum robots for medical applications: A survey," *IEEE Transactions on Robotics*, vol. 31, no. 6, pp. 1261–1280, 2015. 1

[10] R. F. Shepherd, F. Ilievski, W. Choi, S. A. Morin, A. A. Stokes, A. D. Mazzeo, X. Chen, M. Wang, and G. M. Whitesides, "Multigait soft robot," *Proceedings of the national academy of sciences*, vol. 108, no. 51, pp. 20400–20403, 2011. 2, 3

[11] B. S. Homberg, R. K. Katzschmann, M. R. Dogar, and D. Rus, "Haptic identification of objects using a modular soft robotic gripper," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pp. 1698–1705, IEEE, 2015. 2, 3, 16, 17

[12] P. E. Dupont, J. Lock, B. Itkowitz, and E. Butler, "Design and control of concentric-tube robots," *IEEE Transactions on Robotics*, vol. 26, no. 2, pp. 209–225, 2010. 2, 3, 14, 15

[13] S. Yim and M. Sitti, "Design and rolling locomotion of a magnetically actuated soft capsule endoscope," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 183–194, 2012.

[14] T. Kato, I. Okumura, H. Kose, K. Takagi, and N. Hata, "Tendon-driven continuum robot for neuroendoscopy: validation of extended kinematic mapping for hysteresis operation," *International journal of computer assisted radiology and surgery*, vol. 11, no. 4, pp. 589–602, 2016. 2, 3

[15] A. D. Marchese, R. K. Katzschmann, and D. Rus, "A recipe for soft fluidic elastomer robots," *Soft Robotics*, vol. 2, no. 1, pp. 7–25, 2015. 2

[16] A. D. Marchese, C. D. Onal, and D. Rus, "Autonomous soft robotic fish capable of escape maneuvers using fluidic elastomer actuators," *Soft Robotics*, vol. 1, no. 1, pp. 75–87, 2014. 2, 4

[17] K. C. Galloway, P. Polygerinos, C. J. Walsh, and R. J. Wood, "Mechanically programmable bend radius for fiber-reinforced soft actuators," in *Advanced Robotics (ICAR), 2013 16th International Conference on*, pp. 1–6, IEEE, 2013. 2, 4

[18] W. McMahan, V. Chitrakaran, M. Csencsits, D. Dawson, I. D. Walker, B. A. Jones, M. Pritts, D. Dienno, M. Grissom, and C. D. Rahn, "Field trials and testing of the octarm continuum manipulator," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pp. 2336–2341, IEEE, 2006. 2, 4

[19] S. Seok, C. D. Onal, R. Wood, D. Rus, and S. Kim, "Peristaltic locomotion with antagonistic actuators in soft robotics," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pp. 1228–1233, IEEE, 2010. 2, 4

[20] J. Shintake, V. Cacucciolo, H. Shea, and D. Floreano, "Soft biomimetic fish robot made of dielectric elastomer actuators," *Soft robotics*, 2018. 2, 4, 5

[21] H. Wang, W. Chen, X. Yu, T. Deng, X. Wang, and R. Pfeifer, "Visual servo control of cable-driven soft robotic manipulator.," in *IROS*, pp. 57–62, 2013. 2, 4, 5, 11, 13, 14, 15

[22] M. A. Robertson and J. Paik, "New soft robots really suck: Vacuum-powered systems empower diverse capabilities," *Science Robotics*, vol. 2, no. 9, p. eaan6357, 2017. 2, 5

[23] E. W. Hawkes, L. H. Blumenschein, J. D. Greer, and A. M. Okamura, "A soft robot that navigates its environment through growth," *Science Robotics*, vol. 2, no. 8, p. eaan3028, 2017. 2, 5

[24] M. Wehner, R. L. Truby, D. J. Fitzgerald, B. Mosadegh, G. M. Whitesides, J. A. Lewis, and R. J. Wood, "An integrated design and fabrication strategy for entirely soft, autonomous robots," *Nature*, vol. 536, no. 7617, p. 451, 2016. 2, 5

[25] G. Z. Lum, Z. Ye, X. Dong, H. Marvi, O. Erin, W. Hu, and M. Sitti, "Shape-programmable magnetic soft matter," *Proceedings of the National Academy of Sciences*, vol. 113, no. 41, pp. E6007–E6015, 2016. 2, 5

[26] J. C. Case, E. L. White, and R. K. Kramer, "Sensor enabled closed-loop bending control of soft beams," *Smart Materials and Structures*, vol. 25, no. 4, p. 045018, 2016. 3, 14, 15

[27] D.-H. Kim, J.-H. Ahn, W. M. Choi, H.-S. Kim, T.-H. Kim, J. Song, Y. Y. Huang, Z. Liu, C. Lu, and J. A. Rogers, "Stretchable and foldable silicon integrated circuits," *Science*, vol. 320, no. 5875, pp. 507–511, 2008. 3

[28] Y.-L. Park, C. Majidi, R. Kramer, P. Bérard, and R. J. Wood, "Hyperelastic pressure sensing with a liquid-embedded elastomer," *Journal of Micromechanics and Microengineering*, vol. 20, no. 12, p. 125029, 2010. 3

[29] J.-B. Chossat, Y.-L. Park, R. J. Wood, and V. Duchaine, "A soft strain sensor based on ionic and metal liquids," *IEEE Sensors Journal*, vol. 13, no. 9, pp. 3405–3414, 2013. 3

[30] D. J. Lipomi, M. Vosgueritchian, B. C. Tee, S. L. Hellstrom, J. A. Lee, C. H. Fox, and Z. Bao, "Skin-like pressure and strain sensors based on transparent elastic films of carbon nanotubes," *Nature nanotechnology*, vol. 6, no. 12, p. 788, 2011. 3

[31] J. T. Muth, D. M. Vogt, R. L. Truby, Y. Mengüç, D. B. Kolesky, R. J. Wood, and J. A. Lewis, "Embedded 3d printing of strain sensors within highly stretchable elastomers," *Advanced Materials*, vol. 26, no. 36, pp. 6307–6312, 2014. 3

[32] L. Jiang, K. Low, J. Costa, R. J. Black, and Y.-L. Park, "Fiber optically sensorized multi-fingered robotic hand," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pp. 1763–1768, IEEE, 2015. 3, 4, 6

[33] C. To, T. L. Hellebrekers, and Y.-L. Park, "Highly stretchable optical sensors for pressure, strain, and curvature measurement," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pp. 5898–5903, IEEE, 2015. 3

[34] F. Caralt, J. L. Molnar, J. P. Stingel, T. W. Cahoon, and F. L. Hammond, "Diffusion-based optical sensors for multimodal strain measurement in soft devices," in *SENSORS, 2017 IEEE*, pp. 1–3, IEEE, 2017. 3

[35] R. K. Kramer, C. Majidi, R. Sahai, and R. J. Wood, "Soft curvature sensors for joint angle proprioception," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pp. 1919–1926, IEEE, 2011. 3

[36] Y. Mengüç, Y.-L. Park, E. Martinez-Villalpando, P. Aubin, M. Zisook, L. Stirling, R. J. Wood, and C. J. Walsh, "Soft wearable motion sensing suit for lower limb biomechanics measurements," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pp. 5309–5316, IEEE, 2013. 3, 6

[37] J.-B. Chossat, Y. Tao, V. Duchaine, and Y.-L. Park, "Wearable soft artificial skin for hand motion detection with embedded microfluidic strain sensing," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pp. 2568–2573, IEEE, 2015. 3, 6

[38] V. Arabagi, O. Felfoul, A. H. Gosline, R. J. Wood, and P. E. Dupont, "Biocompatible pressure sensing skins for minimally invasive surgical instruments," *IEEE sensors journal*, vol. 16, no. 5, pp. 1294–1303, 2016. 4, 6

[39] K. Noda, E. Iwase, K. Matsumoto, and I. Shimoyama, "Stretchable liquid tactile sensor for robot-joints," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pp. 4212–4217, IEEE, 2010. 4, 6

[40] J. W. Guggenheim, L. P. Jentoft, Y. Tenzer, and R. D. Howe, "Robust and inexpensive six-axis force–torque sensors using mems barometers," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 2, pp. 838–844, 2017. 4, 6

[41] J. Kim, A. Alspach, and K. Yamane, "3d printed soft skin for safe human-robot interaction," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pp. 2419–2425, IEEE, 2015. 4, 6

[42] F. Faure, C. Duriez, H. Delingette, J. Allard, B. Gilles, S. Marchesseau, H. Talbot, H. Courtecuisse, G. Bousquet, I. Peterlik, *et al.*, "Sofa: A multi-model framework for interactive physical simulation," in *Soft Tissue Biomechanical Modeling for Computer Assisted Surgery*, pp. 283–321, Springer, 2012. 5, 77

[43] A. Nealen, M. Müller, R. Keiser, E. Boxerman, and M. Carlson, "Physically based deformable models in computer graphics," in *Computer graphics forum*, vol. 25, pp. 809–836, Wiley Online Library, 2006. 5

[44] M. C. Yip and D. B. Camarillo, "Model-less feedback control of continuum manipulators in constrained environments," *IEEE Transactions on Robotics*, vol. 30, no. 4, pp. 880–889, 2014. 6, 7, 14, 15, 81, 83

[45] C. Duriez, "Control of elastic soft robots based on real-time finite element method," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pp. 3982–3987, IEEE, 2013. 7, 12, 13, 14, 15, 19, 34

[46] F. Largilliere, V. Verona, E. Coevoet, M. Sanz-Lopez, J. Dequidt, and C. Duriez, "Real-time control of soft-robots using asynchronous finite element modeling," in *ICRA 2015*, p. 6, 2015. 15

[47] E. Coevoet, A. Escande, and C. Duriez, "Optimization-based inverse model of soft robots with contact handling," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1413–1419, 2017. 7, 14, 16

[48] J. A. Bell, C. E. Saikus, K. Ratnayaka, V. Wu, M. Sonmez, A. Z. Faranesh, J. H. Colyer, R. J. Lederman, and O. Kocaturk, "A deflectable guiding catheter for real-time mri-guided interventions," *Journal of Magnetic Resonance Imaging*, vol. 35, no. 4, pp. 908–915, 2012. 7

[49] P. M. Loschak, L. J. Brattain, and R. D. Howe, "Algorithms for automatically pointing ultrasound imaging catheters," *IEEE Transactions on Robotics*, vol. 33, no. 1, pp. 81–91, 2017. 7, 16

[50] K. Xu, N. Simaan, *et al.*, "An investigation of the intrinsic force sensing capabilities of continuum robots," *IEEE Transactions on Robotics*, vol. 24, no. 3, pp. 576–587, 2008. 8, 16, 18

[51] C. B. Black, J. Till, and D. C. Rucker, "Parallel continuum robots: Modeling, analysis, and actuation-based force sensing," *IEEE Transactions on Robotics*, vol. 34, no. 1, pp. 29–47, 2018. 8, 16, 18

[52] R. J. Webster III and B. A. Jones, "Design and kinematic modeling of constant curvature continuum robots: A review," *The International Journal of Robotics Research*, vol. 29, no. 13, pp. 1661–1683, 2010. 11, 12, 13

[53] D. Trivedi, A. Lotfi, and C. D. Rahn, "Geometrically exact models for soft robotic manipulators," *IEEE Transactions on Robotics*, vol. 24, no. 4, pp. 773–780, 2008. 12, 13

[54] M. Khoshnam and R. V. Patel, "A pseudo-rigid-body 3r model for a steerable ablation catheter," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pp. 4427–4432, IEEE, 2013. 12, 13

[55] Y. Ganji, F. Janabi-Sharifi, and A. N. Cheema, "Robot-assisted catheter manipulation for intracardiac navigation," *International journal of computer assisted radiology and surgery*, vol. 4, no. 4, pp. 307–315, 2009. 13

[56] D. K. Pai, "Strands: Interactive simulation of thin solids using cosserat models," in *Computer Graphics Forum*, vol. 21, pp. 347–352, Wiley Online Library, 2002. 12, 13

[57] D. Trivedi, A. Lotfi, and C. D. Rahn, "Geometrically exact dynamic models for soft robotic manipulators," in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pp. 1497–1502, IEEE, 2007. 12

[58] W. Tang, T. R. Wan, D. A. Gould, T. How, and N. W. John, "A stable and real-time nonlinear elastic approach to simulating guidewire and catheter insertions based on cosserat rod," *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 8, pp. 2211–2218, 2012. 13

[59] F. Renda, M. Giorelli, M. Calisti, M. Cianchetti, and C. Laschi, "Dynamic model of a multibending soft robot arm driven by cables," *IEEE Transactions on Robotics*, vol. 30, no. 5, pp. 1109–1122, 2014. 12

[60] L. B. Kratchman, T. L. Bruns, J. J. Abbott, and R. J. Webster, "Guiding elastic rods with a robot-manipulated magnet for medical applications," *IEEE Transactions on Robotics*, vol. 33, no. 1, pp. 227–233, 2017. 13

[61] T. Greigarn and M. C. Çavuşoğlu, "Task-space motion planning of mri-actuated catheters for catheter ablation of atrial fibrillation," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pp. 3476–3482, IEEE, 2014. 13

[62] C. Duriez, C. Andriot, and A. Kheddar, "Signorini's contact model for deformable objects in haptic simulations," in *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, vol. 4, pp. 3232–3237, IEEE, 2004. 13, 16, 18, 20, 82

[63] C. Duriez, C. Andriot, and A. Kheddar, "A multi-threaded approach for deformable/rigid contacts with haptic feedback," in *Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2004. HAPTICS'04. Proceedings. 12th International Symposium on*, pp. 272–279, IEEE, 2004. 13

[64] C. Duriez, S. Cotin, J. Lenoir, and P. Neumann, "New approaches to catheter navigation for interventional radiology simulation," *Computer aided surgery*, vol. 11, no. 6, pp. 300–308, 2006. 13, 16, 18

[65] H. Courtecuisse, J. Allard, P. Kerfriden, S. P. Bordas, S. Cotin, and C. Duriez, "Real-time simulation of contact and cutting of heterogeneous soft-tissues," *Medical image analysis*, vol. 18, no. 2, pp. 394–410, 2014. 13, 16, 18

[66] P. Polygerinos, Z. Wang, J. T. Overvelde, K. C. Galloway, R. J. Wood, K. Bertoldi, and C. J. Walsh, "Modeling of soft fiber-reinforced bending actuators," *IEEE Transactions on Robotics*, vol. 31, no. 3, pp. 778–789, 2015. 13, 14, 15

[67] N. W. Bartlett, M. T. Tolley, J. T. Overvelde, J. C. Weaver, B. Mosadegh, K. Bertoldi, G. M. Whitesides, and R. J. Wood, "A 3d-printed, functionally graded soft robot powered by combustion," *Science*, vol. 349, no. 6244, pp. 161–165, 2015. 13

[68] H. Courtecuisse, Y. Adagolodjo, H. Delingette, and C. Duriez, "Haptic rendering of hyperelastic models with friction," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pp. 591–596, IEEE, 2015. 13

[69] T. Liu, N. L. Poirot, D. Franson, N. Seiberlich, M. A. Griswold, and M. C. Çavuşoğlu, "Modeling and validation of the three-dimensional deflection of an mri-compatible magnetically actuated steerable catheter," *IEEE Transactions on Biomedical Engineering*, vol. 63, no. 10, pp. 2142–2154, 2016. 13

[70] A. Elatta, L. P. Gen, F. L. Zhi, Y. Daoyuan, and L. Fei, "An overview of robot calibration," *Information Technology Journal*, vol. 3, no. 1, pp. 74–78, 2004. 14

[71] M. Shah, R. D. Eastman, and T. Hong, "An overview of robot-sensor calibration methods for evaluation of perception systems," in *Proceedings of the Workshop on Performance Metrics for Intelligent Systems*, pp. 15–20, ACM, 2012. 14

[72] R. J. Webster, J. M. Romano, and N. J. Cowan, "Kinematics and calibration of active cannulas," in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pp. 3888–3895, IEEE, 2008. 14

[73] M. Mahvash and P. E. Dupont, "Stiffness control of surgical continuum manipulators," *IEEE Transactions on Robotics*, vol. 27, no. 2, pp. 334–345, 2011. 14

[74] C. Escande, T. Chettibi, R. Merzouki, V. Coelen, and P. M. Pathak, "Kinematic calibration of a multisection bionic manipulator," *IEEE/ASME transactions on mechatronics*, vol. 20, no. 2, pp. 663–674, 2015. 14

[75] N. Tan, X. Gu, and H. Ren, "Simultaneous robot-world, sensor-tip, and kinematics calibration of an underactuated robotic hand with soft fingers," *IEEE Access*, 2017. 14

[76] K. F. Augenstein, B. R. Cowan, I. J. LeGrice, P. M. Nielsen, and A. A. Young, "Method and apparatus for soft tissue material parameter estimation using tissue tagged magnetic resonance imaging," *Journal of biomechanical engineering*, vol. 127, no. 1, pp. 148–157, 2005. 14

[77] B. Frank, R. Schmedding, C. Stachniss, M. Teschner, and W. Burgard, "Learning the elasticity parameters of deformable objects with a manipulation robot," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pp. 1877–1883, IEEE, 2010.

[78] B. Wang, L. Wu, K. Yin, U. Ascher, L. Liu, and H. Huang, "Deformation capture and modeling of soft objects," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 4, p. 94, 2015.

[79] A. Petit, F. Ficuciello, G. A. Fontanelli, L. Villani, and B. Siciliano, "Using physical modeling and rgb-d registration for contact force sensing on deformable objects," in *ICINCO 2017-14th International Conference on Informatics in Control, Automation and Robotics*, vol. 2, pp. 24–33, ScitePress; Springer, 2017. 14, 16, 18, 97

[80] Z. Zhang, A. Petit, J. Dequidt, and C. Duriez, "Calibration and external force sensing for soft robots using an rgb-d camera," *IEEE Robotics and Automation Letters, (under review)*. 14, 16, 27, 57

[81] F. Thérien and J.-S. Plante, "Design and calibration of a soft multiple degree of freedom motion sensor system based on dielectric elastomers," *Soft Robotics*, vol. 3, no. 2, pp. 45–53, 2016. 14

[82] A. D. Marchese, K. Komorowski, C. D. Onal, and D. Rus, "Design and control of a soft and continuously deformable 2d robotic manipulation system," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pp. 2189–2196, IEEE, 2014. 14, 15

[83] A. D. Marchese and D. Rus, "Design, kinematics, and control of a soft spatial fluidic elastomer manipulator," *The International Journal of Robotics Research*, vol. 35, no. 7, pp. 840–869, 2016. 14, 15

[84] A. A. Alqumsan, S. Khoo, and M. Norton, "Robust control of continuum robots using cosserat rod theory," *Mechanism and Machine Theory*, vol. 131, pp. 48–61, 2019. 14, 15

[85] Z. Zhang, J. Dequidt, A. Kruszewski, F. Largilliere, and C. Duriez, "Kinematic modeling and observer based control of soft robot using real-time finite element method," in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pp. 5509–5514, IEEE, 2016. 14, 16, 27, 88

[86] Z. Zhang, T. M. Bieze, J. Dequidt, A. Kruszewski, and C. Duriez, "Visual servoing control of soft robots based on finite element model," in *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*, pp. 2895–2901, IEEE, 2017. 14, 16, 27

[87] J. Back, L. Lindenroth, K. Rhode, and H. Liu, "Model-free position control for cardiac ablation catheter steering using electromagnetic position tracking and tension feedback," *Frontiers in Robotics and AI*, vol. 4, p. 17, 2017. 14, 16

[88] M. C.-S. Yuen, T. R. Lear, H. Tonoyan, M. Telleria, and R. Kramer-Bottiglio, "Toward closed-loop control of pneumatic grippers during pack-and-deploy operations," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1402–1409, 2018. 14, 15

[89] M. C. Yip and D. B. Camarillo, "Model-less hybrid position/force control: A minimalist approach for continuum manipulators in unknown, constrained environments.," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 844–851, 2016. 14, 15

[90] M. C. Yip, J. A. Sganga, and D. B. Camarillo, "Autonomous control of continuum robot manipulators for complex cardiac ablation tasks," *Journal of Medical Robotics Research*, vol. 2, no. 01, p. 1750002, 2017. 14, 16

[91] K.-H. Lee, D. K. Fu, M. C. Leong, M. Chow, H.-C. Fu, K. Althoefer, K. Y. Sze, C.-K. Yeung, and K.-W. Kwok, "Nonparametric online learning control for soft continuum robot: An enabling technique for effective endoscopic navigation," *Soft robotics*, vol. 4, no. 4, pp. 324–337, 2017. 14, 15

[92] T. George Thuruthel, E. Falotico, M. Manti, A. Pratesi, M. Cianchetti, and C. Laschi, "Learning closed loop kinematic controllers for continuum manipulators in unstructured environments," *Soft robotics*, vol. 4, no. 3, pp. 285–296, 2017. 14, 15

[93] V. Vikas, P. Grover, and B. Trimmer, "Model-free control framework for multi-limb soft robots," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pp. 1111–1116, IEEE, 2015. 14, 15

[94] X. You, Y. Zhang, X. Chen, X. Liu, Z. Wang, H. Jiang, and X. Chen, "Model-free control for soft manipulators based on reinforcement learning," in *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*, pp. 2909–2915, IEEE, 2017. 14, 15

[95] O. Kermorgant and F. Chaumette, "Dealing with constraints in sensor-based robot control," *IEEE Transactions on Robotics*, vol. 30, no. 1, pp. 244–257, 2014. 15

[96] N. Cazy, P.-B. Wieber, P. R. Giordano, and F. Chaumette, "Visual servoing when visual information is missing: Experimental comparison of visual feature prediction schemes," in *ICRA'15-IEEE International Conference on Robotics and Automation*, pp. 6031–6036, IEEE, 2015. 15

[97] Y. Bailly, Y. Amirat, and G. Fried, "Modeling and control of a continuum style microrobot for endovascular surgery," *IEEE Transactions on Robotics*, vol. 27, no. 5, pp. 1024–1030, 2011. 16

[98] M. Posa, C. Cantu, and R. Tedrake, "A direct method for trajectory optimization of rigid bodies through contact," *The International Journal of Robotics Research*, vol. 33, no. 1, pp. 69–81, 2014. 16

[99] H. Zhao, K. O'Brien, S. Li, and R. F. Shepherd, "Optoelectronically innervated soft prosthetic hand via stretchable optical waveguides," *Science Robotics*, vol. 1, no. 1, p. eaai7529, 2016. 16, 17

[100] B. Shih, D. Drotman, C. Christianson, Z. Huo, R. White, H. I. Christensen, and M. T. Tolley, "Custom soft robotic gripper sensor skins for haptic object visualization," in *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*, pp. 494–501, IEEE, 2017. 17

[101] R. L. Truby, M. Wehner, A. K. Grosskopf, D. M. Vogt, S. G. Uzel, R. J. Wood, and J. A. Lewis, "Soft somatosensitive actuators via embedded 3d printing," *Advanced Materials*, vol. 30, no. 15, p. 1706383, 2018. 16, 17

[102] P. Polygerinos, A. Ataollahi, T. Schaeffter, R. Razavi, L. D. Seneviratne, and K. Althoefer, "Mri-compatible intensity-modulated force sensor for cardiac catheterization procedures," *IEEE Transactions on biomedical engineering*, vol. 58, no. 3, pp. 721–726, 2011. 16, 17

[103] S. B. Kesner and R. D. Howe, "Robotic catheter cardiac ablation combining ultrasound guidance and force control," *The International Journal of Robotics Research*, vol. 33, no. 4, pp. 631–644, 2014. 17

[104] R. Xu, A. Yurkewich, and R. V. Patel, "Curvature, torsion, and force sensing in continuum robots using helically wrapped fbg sensors," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 1052–1059, 2016. 17

[105] F. Khan, R. J. Roesthuis, and S. Misra, "Force sensing in continuum manipulators using fiber bragg grating sensors," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2017*, 2017. 16, 17

[106] D. C. Rucker and R. J. Webster, "Deflection-based force sensing for continuum robots: A probabilistic approach," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pp. 3764–3769, IEEE, 2011. 16, 18

[107] A. Bajo and N. Simaan, "Kinematics-based detection and localization of contacts along multisegment continuum robots," *IEEE Transactions on Robotics*, vol. 28, no. 2, pp. 291–302, 2012. 18

[108] H. Yuan, P. W. Y. Chiu, and Z. Li, "Shape-reconstruction-based force sensing method for continuum surgical robots with large deformation," *IEEE Robotics and Automation Letters*, vol. 2, no. 4, pp. 1972–1979, 2017. 16, 18

[109] J. Back, T. Manwell, R. Karim, K. Rhode, K. Althoefer, and H. Liu, "Catheter contact force estimation from shape detection using a real-time cosserat rod model," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pp. 2037–2042, IEEE, 2015. 16, 18

[110] M. Khoshnam, A. C. Skanes, and R. V. Patel, "Modeling and estimation of tip contact force for steerable ablation catheters," *IEEE Transactions on Biomedical Engineering*, vol. 62, no. 5, pp. 1404–1415, 2015. 17, 18

[111] M. Kouh Soltani, S. Khanmohammadi, and F. Ghalichi, "A three-dimensional shape-based force and stiffness-sensing platform for tendon-driven catheters," *Sensors*, vol. 16, no. 7, p. 990, 2016. 18

[112] S. Hasanzadeh and F. Janabi-Sharifi, "Model-based force estimation for intracardiac catheters," *IEEE/ASME Transactions on Mechatronics*, vol. 21, no. 1, pp. 154–162, 2016. 16, 18

[113] E. Magrini, F. Flacco, and A. De Luca, "Estimation of contact forces using a virtual force sensor," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pp. 2126–2133, IEEE, 2014. 16, 18

[114] T.-H. Pham, A. Kheddar, A. Qammaz, and A. A. Argyros, "Towards force sensing from vision: Observing hand-object interactions to infer manipulation forces," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2810–2819, 2015. 16, 18

[115] Z. Zhang, J. Dequidt, J. Back, H. Liu, and C. Duriez, "Motion control of cable-driven continuum catheter robot through contacts," *IEEE Robotics and Automation Letters, (under review)*. 16, 81

[116] P. Puangmali, H. Liu, L. D. Seneviratne, P. Dasgupta, and K. Althoefer, "Miniature 3-axis distal force sensor for minimally invasive surgical palpation," *IEEE/ASME Transactions On Mechatronics*, vol. 17, no. 4, pp. 646–656, 2012. 16

[117] U. Kim, D.-H. Lee, W. J. Yoon, B. Hannaford, and H. R. Choi, "Force sensor integrated surgical forceps for minimally invasive robotic surgery," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1214–1224, 2015. 16

[118] A. Colomé, D. Pardo, G. Alenya, and C. Torras, "External force estimation during compliant robot manipulation," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pp. 3535–3540, IEEE, 2013. 18

[119] A. Wahrburg, S. Zeiss, B. Matthias, and H. Ding, "Contact force estimation for robotic assembly using motor torques," in *Automation Science and Engineering (CASE), 2014 IEEE International Conference on*, pp. 1252–1257, IEEE, 2014. 18

[120] M. Lin and S. Gottschalk, "Collision detection between geometric models: A survey," in *Proc. of IMA conference on mathematics of surfaces*, vol. 1, pp. 602–608, 1998. 20

[121] A. Signorini, "Sopra alcune questioni di statica dei sistemi continui," *Annali della Scuola Normale Superiore di Pisa-Classe di Scienze*, vol. 2, no. 2, pp. 231–251, 1933. 20

[122] R. Cottle, "Js pang, and re stone. the linear complementarity problem," 1992. 20

[123] J.-J. E. Slotine, W. Li, *et al.*, *Applied nonlinear control*, vol. 199. Prentice hall Englewood Cliffs, NJ, 1991. 22

[124] F. Marini and B. Walczak, "Particle swarm optimization (pso). a tutorial," *Chemometrics and Intelligent Laboratory Systems*, vol. 149, pp. 153–165, 2015. 31

[125] R. Poli, "Analysis of the publications on the applications of particle swarm optimisation," *Journal of Artificial Evolution and Applications*, vol. 2008, 2008. 31

[126] J. Xin, G. Chen, and Y. Hai, "A particle swarm optimizer with multi-stage linearly-decreasing inertia weight," in *Computational Sciences and Optimization, 2009. CSO 2009. International Joint Conference on*, vol. 1, pp. 505–508, IEEE, 2009. 31

[127] E. Coevoet, T. Morales-Bieze, F. Largilliere, Z. Zhang, M. Thieffry, M. Sanz-Lopez, B. Carrez, D. Marchal, O. Goury, J. Dequidt, *et al.*, "Software toolkit for modeling, simulation, and control of soft robots," *Advanced Robotics*, vol. 31, no. 22, pp. 1208–1224, 2017. 34, 77, 90

[128] H. Lin and P. J. Antsaklis, "Stability and stabilizability of switched linear systems: a survey of recent results," *IEEE Transactions on Automatic control*, vol. 54, no. 2, pp. 308–322, 2009. 41

[129] Z. Zhang, J. Dequidt, and C. Duriez, "Vision-based sensing of external forces acting on soft robots using finite element method," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1529–1536, 2018. 57, 66

[130] H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock, and M. Diehl, "qpoases: A parametric active-set algorithm for quadratic programming," *Mathematical Programming Computation*, vol. 6, no. 4, pp. 327–363, 2014. 60, 86

[131] R. B. Rusu and S. Cousins, "3d is here: Point cloud library (pcl)," in *Robotics and automation (ICRA), 2011 IEEE International Conference on*, pp. 1–4, IEEE, 2011. 63, 64, 65, 78

[132] J. Wang and Y. Yagi, "Integrating color and shape-texture features for adaptive real-time object tracking," *IEEE Transactions on Image Processing*, vol. 17, no. 2, pp. 235–240, 2008. 73

[133] C. Yang, R. Duraiswami, and L. Davis, "Fast multiple object tracking via a hierarchical particle filter," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 1, pp. 212–219, IEEE, 2005. 73

[134] A. Petit, V. Lippiello, G. A. Fontanelli, and B. Siciliano, "Tracking elastic deformable objects with an rgb-d sensor for a pizza chef robot," *Robotics and Autonomous Systems*, vol. 88, pp. 187–201, 2017. 73

[135] J. S. Przemieniecki, *Theory of matrix structural analysis*. Courier Corporation, 1985. 82

[136] L. Chen, D. Goldfarb, *et al.*, "An active-set method for mathematical programs with linear complementarity constraints," *Manuscript, Department of Industrial Engineering and Operations Research, Columbia University*, 2007. 88

[137] L. Bai, J. E. Mitchell, and J.-S. Pang, "On convex quadratic programs with linear complementarity constraints," *Computational Optimization and Applications*, vol. 54, no. 3, pp. 517–554, 2013. 88

[138] J. Back, L. Lindenroth, K. Rhode, and H. Liu, "Three dimensional force estimation for steerable catheters through bi-point tracking," *Sensors and Actuators A: Physical*, 2018. 90

[139] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Sensor Fusion IV: Control Paradigms and Data Structures*, vol. 1611, pp. 586–607, International Society for Optics and Photonics, 1992. 90

[140] O. Goury and C. Duriez, "Fast, generic, and reliable control and simulation of soft robots using model order reduction," *IEEE Transactions on Robotics*, no. 99, pp. 1–12, 2018. 101