Modeling, Calibration, and Intelligently-assisted Telemanipulation of Surgical Continuum Robots

By

Long Wang

Dissertation

Submitted to the Faculty of the

Graduate School of Vanderbilt University

in partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

in

Mechanical Engineering

May 10, 2019

Nashville, Tennessee

Approved:

Nabil Simaan, Ph.D.

Eric Barth, Ph.D.

Thomas Withrow, Ph.D.

Howie Choset, Ph.D.

Pietro Valdastri, Ph.D.

DEDICATION

This dissertation is lovingly dedicated to my wife, Muqun Li, Ph.D. Her support, encouragement, and constant love have sustained me throughout my study.

ACKNOWLEDGMENTS

Firstly, I would like to express my deepest gratitude to my advisor Prof. Nabil Simaan for the continuous support of my graduate study and related research, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my Ph.D. study.

Besides my advisor, I would like to thank the rest of my thesis committee: Prof. Eric Barth, Prof. Thomas Withrow, Prof. Howie Choset, and Prof. Pietro Valdastri, for their insightful comments and encouragement, and for the critical questions which incented me to widen my research from various perspectives. I very much appreciate the mentorship and guidance of Prof. Russ Taylor and Prof. Peter Kazanzides during the research collaboration. Thanks also to Dr. Karen Joos, Dr. Keith Obstein, Dr. James Netterville, and Dr. Kelly Groom for their expertise and assistance on clinical aspects.

I must also acknowledge my labmates at ARMA. Special thanks to Dr. Andrea Bajo, Dr. Rajarshi Roy, Mr. Rashid Yasin, and Mr. Giuseppe Del Giudice, for the countless astute discussions on theoretical and technical aspects that move my research agenda forward. Many thanks to Dr. Jason Pile and Dr. Haoran Yu for their help during my starting days at ARMA.

I am also grateful to colleagues at Carnegie Mellon University and Johns Hopkins University. I had great pleasure working with Dr. Arun Srivatsan Rangaprasad and Dr. Elif Ayvali, who have been and will always be inspirations to me. Particularly helpful to me during the research collaboration were Dr. Zihan Chen, Dr. Preetham chalasani, and Mr. Anton Deguet, who provided crucial assistance in software architecture and implementation.

Last but not the least, this dissertation would not have been possible without the support of my family. I'm extremely grateful to my wife and parents.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

1.1 Background and Motivation

Continuum robots allow deep access to the anatomy while providing an inherent safety owing to their structural compliance. They have gained great interests from both academia and industry for the past decade. However, until now, one of the biggest challenges that limit the success of these robots has been the extra hurdles in achieving accurate control and easy telemanipulation. Compared to traditional rigid laparoscopic surgical instruments, additional difficulties stem from their counter-intuitive snake-like structure, modeling uncertainties due to their compliant mechanisms that lead to control challenges, and the complex interaction between the entire flexible robot body and the anatomy.

To overcome this barrier and to unlock the full potential of surgical continuum robots, this dissertation introduces two ways to alleviate the aforementioned obstacles. First, an intelligent assistive telemanipulation framework is proposed for surgeon-robot interaction, where virtual fixture laws are used and updated based on sensory feedback during surgery. Second, a rectified kinematics model that enables towards semi-automation is calibrated to reduce the burden on surgeons in learning and compensating for robot modeling errors. We believe that the potential success of surgical continuum robots benefits significantly future surgical approaches such as single port access surgery and natural orifice transluminal endoscopic surgery. We envision that these robots in the future will be able to provide safe deep access with surgical sub-task semi-automation capabilities and accurate control despite their complex mechanical architecture.

1.2 Continuum Robots for Surgery

Continuum robots have found increased applications in minimally invasive surgery [11]. These applications currently rely on telemanipulation under visual supervision of

a user. Future applications of these robots may involve semi-automated or automated task execution (e.g. automated suturing [83]). The success of such semi-automated and automated tasks depends on the availability of exact kinematic and static models of these robots.

There have been works aiming at improving the accuracy of the kinematic models of continuum robots, investigating modeling approaches [17, 18, 111, 112, 33, 51, 62, 78, 75], and calibration or experimental characterization [99, 12, 27, 64, 41]. A detailed literature review can be found in Chapter 3, from which we observe that there are very sparse works on calibration of continuum robots and that these works do not provide a generalizable formalism that can account for a wide range of continuum robot architectures, or use highly restrictive assumptions regarding bending shapes of these robots, or lack of torsional twist. Furthermore, a formulation for the calibration identification Jacobians, which is critical for the study of appropriate selection of calibration configurations and sensor placement, is generally lacking from continuum robot literature. This work aims to fill this scientific gap and uses the multi-backbone continuum robot architecture as an example to demonstrate the strength of the calibration formulation presented in Chapter 3.

1.3 Technical Gaps and Key Contributions

Using surgical continuum robots as smart slave robots in an intelligent framework can improve the surgical performance and provide safer interaction with environment. This dissertation builds upon previous works on intelligent continuum robots including the intrinsic force sensing in [102, 105], the contact detection and estimation [7, 8], the compliant motion [34, 36] and the hybrid force position control [9]. This dissertation aims to propose an intelligently-assisted framework for surgical continuum robots and to identify and solve the technical challenges in integrating such robots to the framework. Hence we aim to bridge the technical gaps summarized as below, and the specific technical merit and significance are explained in details following the summary.

1. Recent works have shown how robots may used for palpation and probing. However, to date there are limited works using information from force-controlled exploration to update the geometry of the pre-operative surgical plan or model. We will address this gap by exploring methods of robotic force-controlled exploration. In chapter 2 we present a contribution in using robotic force-controlled exploration to update virtual fixture geometry for model-mediated telemanipulation.

2. Within the context of using continuum robots for exploration and model update or semi-automated task execution, the need for an accurate kinematic model of these robots arises. The literature in this area lacks formulations for kinematic calibration of continuum robots. Problems of observability and the proper way of carrying out calibration (e.g. kinematic mapping between which spaces to use) are not investigated. The effect of sensor placement and the number of sensors needed have also not been explored. In chapter 3 we present a contribution in formulating and developing a framework for geometric calibration relying on the identification Jacobians and sensor kinematic measurements.

3. A recent work from our group [21] demonstrates that continuum robots can also carry out micro-scale motions by modulating its equilibrium, which can be used for applications involving imaging-based exploration of the environment (e.g. optical coherence tomography of a patch of tissue). There is a lack of models for control of continuum robots with equilibrium modulation. In this thesis we focus on formulating a framework of simplified kinematics via coupling moment effects and model calibration. In chapter 4 we present such kinematic framework.

1.3.1 Contribution I - Model-mediated Telemanipulation & Virtual Fixture Update using Force-controlled Exploration

During robot-assisted and computer-aided surgery, surgeons attempting to carry out path following tasks such as ablation or dissection along a desired anatomical path are

challenged by the flexibility of the underlying anatomy. Examples of this task can be found in cardiac ablation for electrophysiology and in cholecystectomy where dissection to expose the hepatic and cystic ducts are required. The introduction of image-guided surgery assists surgeons in avoiding critical anatomical structures. In addition, robot assisted image-guided surgery improves the coupling between surgical pre-planning and surgical execution. The success of this coupling hinges on successful registration between the *a-priori* model of the surgical plan and the anatomy as obtained from pre-operative imaging. One key challenge to the paradigm of image-guided surgery is the fact that flexible organs are susceptible to deformation due to gravitational forces or changes in their boundary conditions when the connective tissues around the target organ are removed or displaced to gain access to that organ.

As an initial demonstration of the intelligently-assisted framework, the problem is designed as follows. Given an *a-priori* model and an associated telemanipulation virtual fixture descriptor, the framework allows force-controlled slave robot exploration and correction on the *a-priori* model for deformation and registration errors using the explored data. Then the updated virtual fixture geometry is used in a model-mediated telemanipulation for force-controlled ablation.

Two key contributions are presented in this part. First, a method for force-controlled telemanipulated exploration is proposed to collect geometric data of the deformed environment. With the exploration data, an *a-priori* model of the environment is registered and corrected using deformable registration based on coherent point drift [65]. Using this approach, a flexible environment model is updated for deformation and registration errors.

The second contribution is technical in nature and is the presentation of a highly modularized framework of system integration using the *cisst* package and Simulink® Real-Time. This framework provides assistive virtual fixtures on the master side while supporting model-mediated telemanipulation. It also provides intelligent control behaviors on the slave side to support force-controlled exploration and telemanipulation.

1.3.2 Contribution II - Geometric Calibration of Continuum Robots

Continuum robots use flexible members to achieve complex three dimensional static equilibrium shapes [74, 38]. The kinetostatic behavior of these robots depend on material properties and structural characteristics such as friction, stress concentrations and cross-sectional flexural rigidity. Continuum robots can be categorized into three mechanical architectures: wire actuated single backbone robots (e.g., [37]), concentric tube single backbone robots (e.g., [23, 100]) and multi-backbone continuum robots (e.g. [102]). Continuum robots have found increased applications in minimally invasive surgery [11] and in field robotics [93]. Most of these applications currently rely on telemanipulation under visual supervision of a user. Future applications of these robots may involve semi-automated or automated task execution (e.g. automated suturing [83]). Hence the problem of kinematic calibration for such robots needs to be addressed in this dissertation. The problem of kinematic calibration arises from the fact that future applications of these robots may involve semi-automated or automated task execution (e.g. automated suturing [83]).

The contribution of this part is in presenting a kinematic modeling framework for multi-backbone continuum robots that captures bending shape deviation and coupled twisting and bending of these robots as a result of imperfect assembly. We put forth the concept of bending shape homotopies as a means of parameterizing shape deviations using modal representations inspired by [18] and we derive identification jacobian. Using a single port access system as a validation platform, we validate the kinematic calibration framework experimentally while augmenting these results with additional simulation validations.

1.3.3 Contribution III - Modeling of Continuum Robots with Equilibrium Modulation

Current robotic slave arms for minimally invasive surgery (MIS) offer excellent distal dexterity for surgical tasks requiring large workspaces and position accuracy ranging between 0.5 to 1.5 mm. For example, Kwartowitz et al. [52, 54], experimentally determined the root mean square (RMS) localization accuracy of the da-Vinci Classic and da-Vinci S

to be 1.02 mm and 1.05 mm, respectively. Despite their accuracy errors, these robots are able to perform well when telemanipulated since their repeatability usually exceeds their accuracy and the users can use visual feedback to overcome accuracy errors. Despite increases in precision, existing surgical systems are unable to traverse curved anatomical passageways to perform micro-surgical interventions and also are not suitable for tasks requiring accuracy below 1 mm.

Continuum Robot with Equilibrium Modulation (CREM) can be used to design a multitude of flexible parallel robots and continuum robots with Multi Scale Motion (MSM) capabilities benefiting microsurgery, micro-assembly, and inspection in tight spaces. This concept was first introduced in [21] which includes parallel architecture robots having flexible legs supporting a common load and snake-like robots with multiple backbones (termed parallel robots with constrained continuum legs).

The contribution of this part is in presenting a simplified model for micro-motion of robots with equilibrium modulation. The approach presented takes advantage of the calibration framework to capture minute changes in the configuration-to-task space mapping while avoiding the use of an exact solution to the minimum energy equilibrium. The framework presented provides both instantaneous kinematics and error prorogation for micro-scale motion, which can be used for calibration and control of the micro-scale motion. The modeling approach along with the calibration method was validated experimentally on a multi-backbone continuum robot.

1.4 Broader Implications of This Work

This dissertation has been part of a synergistic collaboration among research groups at Vanderbilt University, Carnegie Mellon University, and Johns Hopkins University. As a first attempt to use force sensing information during minimally invasive surgery, in Chapter 2, we present a method to update virtual fixture geometry using the contact locations obtained from a force-controlled exploration.

Extending the telemanipulation framework and the force-controlled exploration, there are other collaborative efforts that investigate the use of force and stiffness information, as well as more efficient strategies to explore an unknown environment (anatomy). Srivatsan proposed a complementary model update (CMU) method as a more robust approach for rigid registration using stiffness and geometry information to improve registration [91]. Chalasani used Gaussian processes to simultaneously estimate the stiffness and surface of an organ using continuous palpation motion [13, 14]. Ayvali introduced a Bayesian optimization framework to guide probing to maximize information gain, thus avoid probing the entire organ, while registering the predicted stiffness to an *a-priori* geometric model [5, 4].

Chapter 2

Virtual Fixture Geometry Update Using Force-Controlled Exploration

2.1 Related Work

Researchers have been investigating the use of contact information during interaction with organs. Extraction of stiffness information using mechanical imaging via tactile sensor arrays were demonstrated in [30, 47, 25]. Rolling mechanical imaging was obtained using a force-sensitive probe in [59] and probing motion (indenting tissue in the depth direction) was investigated in [105, 70]. Force controlled telemanipulation of continuum robots was used in [9] to map the geometry and stiffness of the environment. Dynamic excitation of tissue was applied to estimate impedance parameter in [35]. This work differs from these previous works by attempting to use geometric information obtained through force-controlled exploration to register and correct a pre-operative *a-priori* model of the surgical plan.

In this chapter, the use of force-controlled scanning of tissue is explored as a means of gathering information for registering and updating the pre-operative model. The most relevant works on this topic include [81] where constrained Kalman filtering was employed to use the contact and estimated stiffness information to obtain a rigid registration of the model. Similarly, the complementary model update (CMU) method [91] was presented as a more robust approach for rigid registration using stiffness and geometry information to improve registration. In addition, Gaussian processes were used to simultaneously estimate the stiffness and surface of an organ using continuous palpation motion [13, 14]. In [5, 4], a Bayesian optimization framework was introduced to guide probing to maximize information gain, thus avoid probing the entire organ, while registering the predicted stiffness to an *a-priori* geometric model. These works account for local deformation induced during the probing process yet disregard a potential global deformation of the

organ. Global deformation of a model is addressed in [65, 10]. This work complements these efforts by testing a naive approach which relies on a force-controlled scan of the organ and which is coupled with a deformable registration. The method is not time-efficient so we anticipate using it to initialize our registration while subsequently achieving a continuous model update using the other approaches listed above.

2.2 Telemanipulation Framework

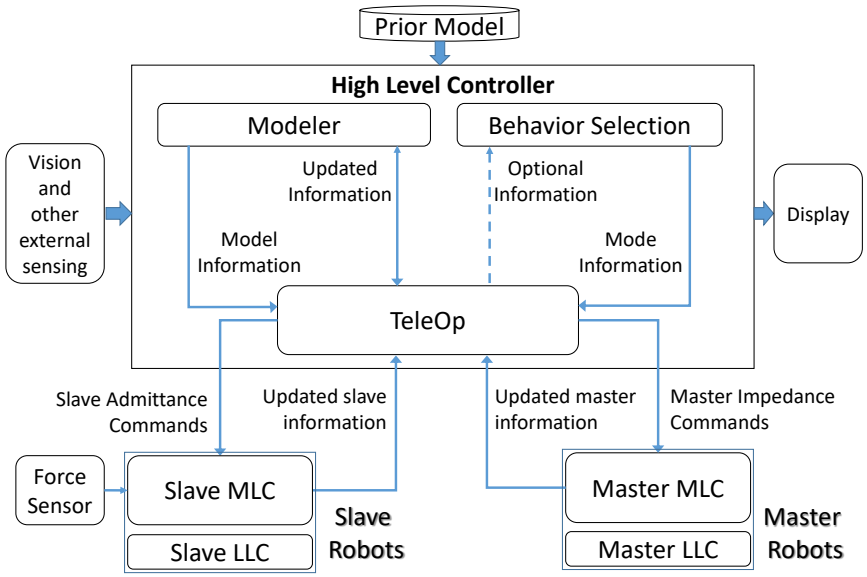


Figure 2.1: System Architecture

Our telemanipulation framework, as shown in Fig. 2.1, is based on the JHU "Surgical Assistant Workstation (SAW)" software environment [42] and the da Vinci Research Toolkit (dVRK) [16, 46]. This environment supports multiple telemanipulation hardware and software components in a mix-and-match fashion. For our current research, we use dVRK master tool manipulators (MTMs) and slave manipulators choosing from either dVRK patient side manipulators (PSMs, as in Fig. 2.3) or a custom Cartesian robot seen in Fig. 2.2. Each slave system manipulates a force-sensing probe comprising either ATI Nano-17 or an ATI Gamma-SI-130-10 Force/Torque sensor with a ball probe finger for contacting tissue, as in Fig. 2.2. In Fig. 2.3, the phantom model is equivalently mounted on an

ATI Gamma Gamma-SI-130-10 Force/Torque sensor for implementation ease to verify the framework of this chapter. The results reported in section 2.5 were obtained with both the Cartesian robot and a dVRK PSM. The component-based SAW software is very modular and its processes may be run on a highly distributed computing environment. Several key processes are discussed below.

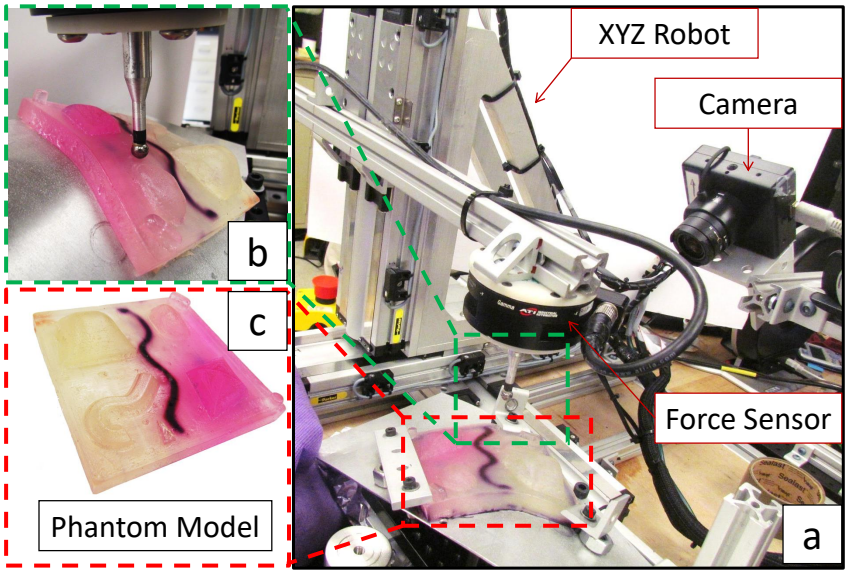


Figure 2.2: A custom Cartesian Slave Robot System: (a) Experiment setup, (b) Ball Probe Finger ATI Force Torque Sensor, (c) A phantom model used in experiment

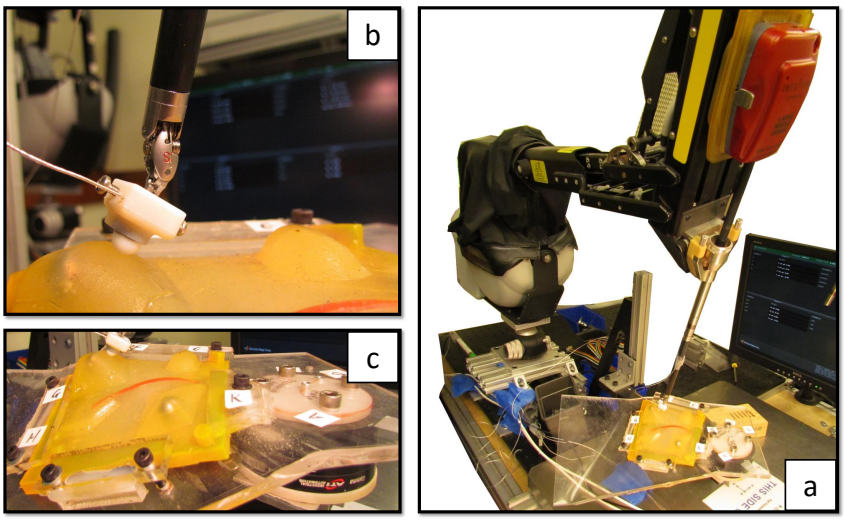


Figure 2.3: PSM Experiment Setup: (a) Experiment setup, (b) Ball probe finger adapter integrated with EM tracker, (c) A phantom model mounted on a force plate

The *Master Controller* process is responsible for the control of the Master manipulator hardware. This process consists of two sub-processes: a *Master Mid-Level Controller (MLC)* which communicates with the *TeleOp* process (described below) and a *Master Low-Level Controller (LLC)* which communicates with the Master hardware and performs basic joint-level servo control functions. The *Master MLC* runs as a clock-driven process at a sampling rate of 500 Hz and the *Master LLC* runs at 1.5 kHz. The *Master MLC* receives impedance specification commands from the *TeleOp* process and translates them into an appropriate form for execution by the *Master LLC*. The *Master MLC* process also returns state information to the *TeleOp* process, including joint and Cartesian positions and velocities, Master gripper openings, and forces and torques exerted by the Master on the surgeon's hand.

The *Slave Controller* process is responsible for control of the Slave hardware. Like the Master Controller, this process consists of a *Slave Mid-Level Controller (MLC)* which communicates with the *TeleOp* process and a *Slave Low-Level Controller (LLC)* which communicates with the Slave hardware. The *Slave MLC* runs as a clock-driven process at a sampling rate of 500 Hz and the *Slave LLC* runs at 1000 Hz. The Slave Controller also contains a force sensing component that reads the slave's force sensor and computes forces exerted on the finger probe. The *Slave MLC* receives admittance commands and virtual fixture specifications from the *TeleOp* process and translates them into Cartesian or joint position/velocity commands that are passed on to the *Slave LLC*. The *Slave MLC* receives state information from the *Slave LLC*, combines this information with other Slave Controller information (e.g., forces, contact information) and passes the combined state information back to the *TeleOp* process.

The *TeleOp* process is the central control point for the system. This process runs as a real-time, clock driven process (at 500 Hz). It is responsible for managing communications among the *Master Controller*, *Slave Controller*, *Modeler*, and higher-level *Behavior Selection* processes. It is also directly responsible for real-time telemanipulation behavior. The

TeleOp process receives state information from the *Master MLC* and *Slave MLC* and passes this information on to the Modeler and the Behavior Selection Process. Based on the entire combined state information (Master, Slave, Model, etc.) and the current behavior mode, the *TeleOp* component determines appropriate admittance commands and sends them to the Slave Controller. Similarly, it also determines appropriate impedance commands and sends them to the *Master Controller*. The *TeleOp* component also has a special "autonomous scanning" behavior in which the slave is issued a series of admittance commands causing the slave manipulator to move at a constant velocity across a surface while exerting regulated force normal to the surface.

The *Behavior Selection* process runs in the background and communicates with the *TeleOp* process to inform it of changes in desired behavior (e.g., simple position following telemanipulation, model-mediated telemanipulation, telemanipulation with force bias, telemanipulation with superimposed palpation motion, etc.). It receives state information from the *TeleOp* process and the Modeler, as well as direct input from the user. It also will manage information displays and other user interfaces not directly involving telemanipulation.

The *Modeler* process is responsible for maintaining a model of the manipulation environment. In the current implementation, the model consists of a triangulated surface mesh representation of an anatomic organ or phantom object. This mesh is augmented with a spline curve representing a path on the surface that the robot is to follow. In future versions, the mesh will also be annotated with stiffness information associated with each triangle in the mesh. In our prior work [35] we demonstrated force-controlled estimation of flexible environment constraints and impedances. In [92, 81] we adapted the constrained extended Kalman filter to allow taking into account geometric and stiffness information to benefit registration in flexible environments. In this work we are extending these results to demonstrate the utility of force-controlled exploration for updating the model. The *Modeler* process also is responsible for maintaining the registration between the slave robot and the model, based on surface contact information provided by the Slave process.

2.2.1 Slave Robot Controllers

To enable hybrid force/motion capabilities, two slightly different versions of Mid-Level Controllers are developed in Matlab Simulink® Real-Time™environment, for a dVRK PSM and a custom Cartesian robot, shown as in Fig. 2.4 and Fig. 2.5, respectively. The dVRK PSM *MLC-LLC* is implemented as an admittance type controller while the Cartesian robot is implemented as a hybrid motion/force with dynamic compensation controller. The motion/force projection mechanisms and the Cartesian robot controller are both motivated by the works of Khatib [48] and Featherstone [28].

The MLC accepts commands from *TeleOp* in a format either of position, admittance, or desired force. A velocity command $\dot{\mathbf{x}}_{des}$ is generated based on a resolved rates algorithm[101], given the desired position $\mathbf{x}_{des}$ and the current slave position. Another velocity command $\dot{\mathbf{x}}_{adm}$ is generated given an admittance force command (or force error in PSM's case) and admittance gains. In the case of PSM controller (Fig. 2.4), $\dot{\mathbf{x}}_{des}$ and $\dot{\mathbf{x}}_{adm}$ are decomposed by the *Motion Force Projection* block, resulting in $\dot{\mathbf{x}}_{des_\perp}$ and $\dot{\mathbf{x}}_{adm_\perp}$ respectively. The added velocity command $\dot{\mathbf{x}}_{cmd}$ is then sent to the PSM LLC. In the case of Vanderbilt Cartesian robot controller, $\dot{\mathbf{x}}_{des}$ and $\dot{\mathbf{x}}_{adm}$ are first combined to be the motion command $\dot{\mathbf{x}}_{cmd}$. This motion command $\dot{\mathbf{x}}_{cmd}$ and the desired force $\mathbf{f}_{des}$ are decomposed by the *Motion Force Projection* block. The projected velocity and force command $\dot{\mathbf{x}}_{cmd_\perp}$ and $\mathbf{f}_{cmd_\perp}$ are then sent to the LLC of the Cartesian robot. The projection matrices in Fig. (2.4, 2.5) are given as follows [48, 28].

$$\begin{aligned} \boldsymbol{\Omega} &= \mathbf{N}(\mathbf{N}^T\mathbf{N})^{-1}\mathbf{N}^T = \mathbf{I} - \bar{\boldsymbol{\Omega}}, \\ \bar{\boldsymbol{\Omega}} &= \mathbf{T}(\mathbf{T}^T\mathbf{T})^{-1}\mathbf{T}^T = \mathbf{I} - \boldsymbol{\Omega}, \\ \mathbf{N} &\in \mathbb{R}^{m\times r}, \mathbf{T} \in \mathbb{R}^{m\times(m-r)} \end{aligned} \tag{2.1}$$

where m is the total task space dimension and r is the force/torque controlled space dimension, in our research $m = 3, r = 1$. As a result, $\mathbf{N} = \mathbf{n}_d = [n_x, n_y, n_z]^T$ specifies the *desired* force

control direction.

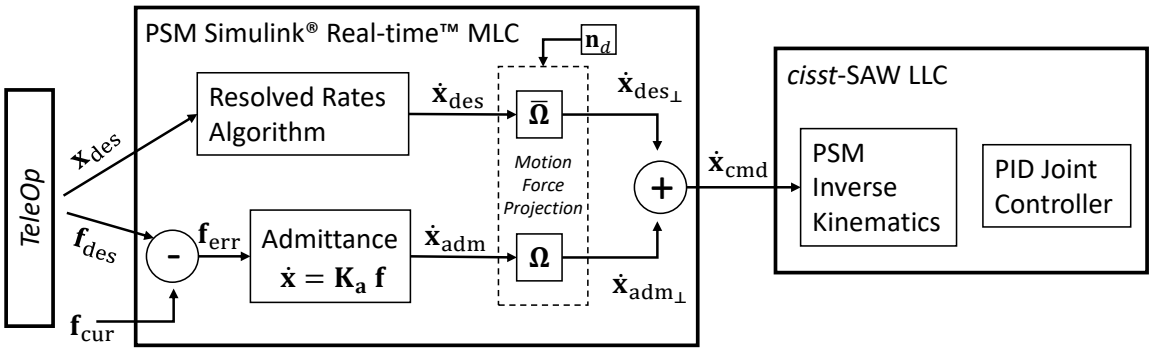


Figure 2.4: PSM Force-controlled Slave MLC-LLC Controller

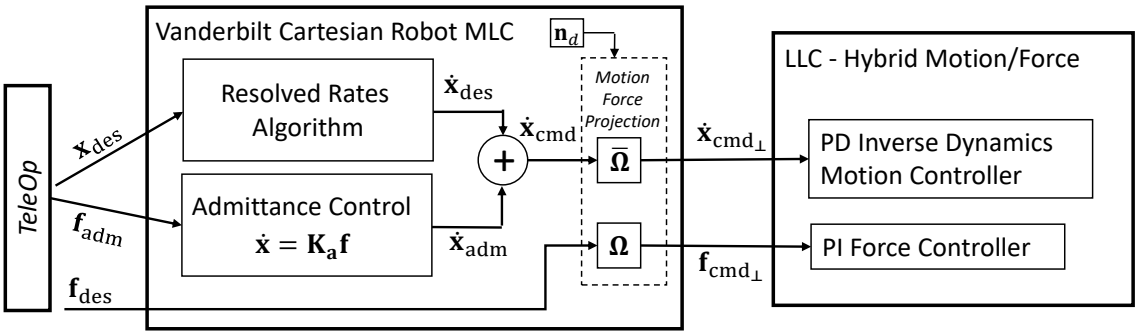


Figure 2.5: Vanderbilt Slave MLC-LLC Controller

2.2.2 Master Impedance Controller

The MLC is implemented as an impedance type controller, which allows combining different control goals by simply adding desired joint torques computed separately. As shown in Fig. 2.6, gravity compensation is rendered at any time and an impedance type virtual fixture controller is running in parallel, taking commands from the teleoperation component.

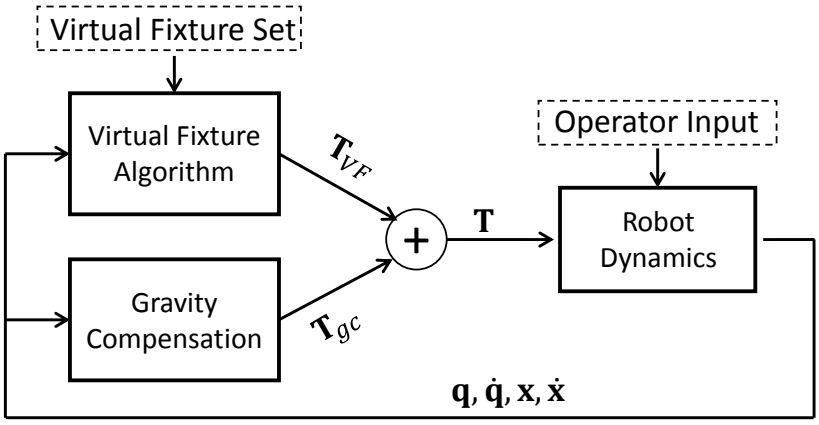


Figure 2.6: Master MLC Impedance Type Controller:
q - joint position; $\dot{\mathbf{q}}$ - joint velocity; **x** - cartesian position; $\dot{\mathbf{x}}$ - cartesian velocity; T - total joint torque applied to robot; T_{VF} - joint torque from virtual fixture controller T_{gc} - joint torque from gravity compensation

To define the virtual fixture controller behavior, *TeleOp* sets force position compliance frame $\mathbf{F}_c = [\mathbf{R}_c, \mathbf{p}_c]$ defined in master base frame. The virtual fixture law also uses position stiffness gain vectors $\mathbf{k}^{(+)}, \mathbf{k}^{(-)}$, position damping gain vectors $\mathbf{b}^{(+)}, \mathbf{b}^{(-)}$ and force bias terms $\mathbf{a}^{(+)}, \mathbf{a}^{(-)}$. The pairs are used to distinguish between movement toward the virtual fixture v.s. away from the virtual fixture boundary. Algorithm 1 shows how the desired force applied on the master tip is computed.

One advantage of this design is that it permits very fast haptic rendering of discontinuous impendence environments when the slave end effector is near the virtual fixture boundary, such as encountered when one is palpating or following an organ surface. It also permits very versatile descriptions of local virtual fixtures behavior, such as encountered in curve following. Further, it permits simple combinations of virtual fixture elements, such as combining surface following with curve following. It is simple to implement and provides a versatile command interface between the *TeleOp* process and the Master Controller.

Algorithm 1 Master Virtual Fixtures Controller

Given

$\mathbf{F} = [\mathbf{R}, \mathbf{p}]$: current pose $\dot{\mathbf{p}}$: current velocity ▷ $\mathbf{F} \in SE(3)$

$\mathbf{F}_c = [\mathbf{R}_c, \mathbf{p}_c]$: position compliance frame w.r.t master

$\mathbf{k}^{(+)}, \mathbf{k}^{(-)}$: stiffness gains $\mathbf{b}^{(+)}, \mathbf{b}^{(-)}$: damping gains

$\mathbf{a}^{(+)}, \mathbf{a}^{(-)}$: force bias terms ▷ $\mathbf{k}, \mathbf{b}, \mathbf{a} \in \mathbb{R}^3$

Compute

if (Enabled) **then**

  $\mathbf{e} = \mathbf{F}_c^{-1}\mathbf{p} = \mathbf{R}_c(\mathbf{p} - \mathbf{p}_c)$ ▷ position error

  ${}^c\mathbf{v} = \mathbf{R}_c^{-1}\dot{\mathbf{p}}$ ▷ velocity written in $\{C\}$

  for $i \in \{x, y, z\}$ **do** ▷ each component

    if $(e_i \leq 0)$ **then** $g_i = a_i^{(-)} + k_i^{(-)} e_i + b_i^{(-)c} v_i$

    else $g_i = a_i^{(+)} + k_i^{(+)} e_i + b_i^{(+)c} v_i$

    end if ▷ gains selection depends on error sign

  end for

  $\mathbf{g} = [g_x, g_y, g_z]^T$ ▷ virtual fixture force in $\{C\}$

  $\boldsymbol{\tau} = \mathbf{R}_c\mathbf{g}$ ▷ virtual fixture force in master base frame

end if

2.2.3 Model-based Virtual Fixtures for Surface Following, Palpation, and Surface Feature Tracking

Although the Slave Controller is capable of implementing virtual fixtures incorporating both positional and force constraints using the methods described in [44, 56], for the current chapter, we rely on impedance commands to exert feedback forces on the surgeon's hands with Master manipulator, based on the current registered model. To simplify the discussion we will treat the Master, Slave, and Model coordinate systems as equivalent, i.e., a position $\vec{p}$ in the Master manipulator coordinates corresponds to position $\vec{p}$ in the Slave and Model. Thus, we will say that the Master is "in contact" with the Model if its current position $\vec{p}$ is on or below the surface of the Model.

For *surface following*, our goal is to exert a constant force normal to the surface while

permitting the surgeon to move the robot freely across it. In this mode, *TeleOp* determines the closest point on the surface from the Master manipulator. This closest point is chosen as the origin of the compliance reference frame, as shown in Fig. 2.7, along with the surface normal at this point as positive Z axis with 0 positive gain and large negative gain. In the mean while, the X and Y axes can be chosen freely, with 0 gains, since motion along the surface is not limited.

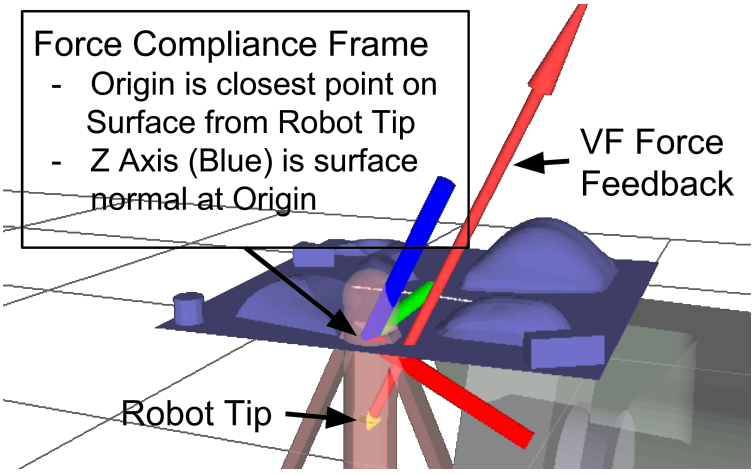


Figure 2.7: Surface following frame with master robot tip and force feedback

For *surface feature tracking* our goal is to assist the surgeon in tracing a predefined curve across the surface while still maintaining contact with the surface with a constant normal force. Along with the surface following virtual fixture, a preregistered curve guidance virtual fixture also starts rendering whenever the robot is close to the curve. Once started, *Teleop* then determines the closest point on the curve and the tangent direction of the curve at the closest point. Similarly, the surface normal at the closest point is picked as the positive Z axis of the compliance reference frame. The tangent direction serves as the X axis with zero gains. The Y axis is determined from the X and Z axes, with large gains for positive and negative directions, as shown in Fig. 2.8. The use of the model-mediated virtual fixture is demonstrated in *Multimedia Extension I*[1].

[1] http://arma.vuse.vanderbilt.edu/images/stories/videos/long_jmr_mme1.mp4

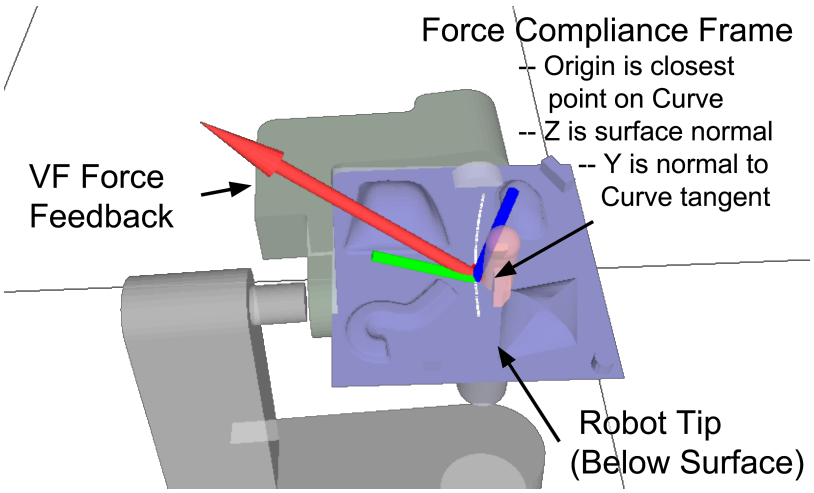


Figure 2.8: Curve following frame with master robot tip and force feedback

2.3 Force Controlled Exploration

Using hybrid force/motion controller of the slave robots described in section 2.2 we can achieve a force-controlled exploration of the environment, i.e. a *surface following* function. Our previous work [35] used a similar *surface following* function where a Cartesian robot was controlled given a constant predefined force desired direction (i.e. $\mathbf{n}_d$ in Fig. 2.5 is constant and specified). In this work, the force desired direction is updated based on current estimation of the environment and robots with wrist orientation control are considered. The surface geometry is estimated and used as data for registration, which will be discussed in section 2.4. The exploration control strategy is described in Fig. 2.9. The hybrid force/motion slave controller accepts the position command $\mathbf{x}_{des}$ from either user command or path planning and the force regulating direction $\hat{\mathbf{n}}$ is set to be the current estimated contact surface normal.

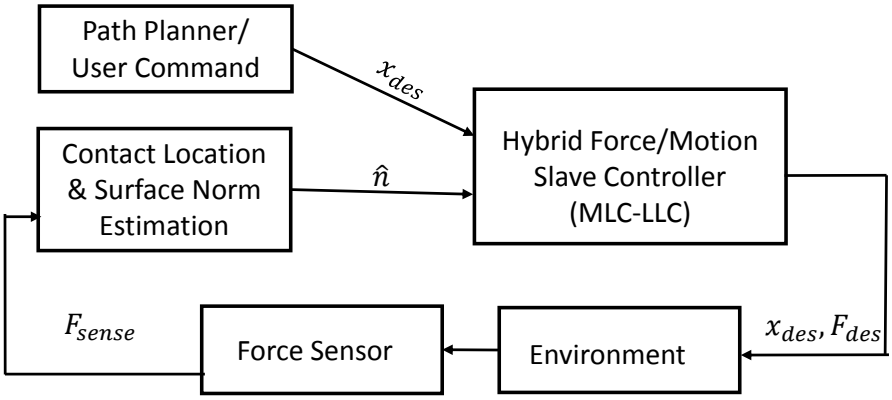


Figure 2.9: Force-controlled Exploration Strategy

2.3.1 Contact and Surface Normal Estimation

The contact location and surface normal estimation is shown in Fig. 2.10. The surface normal is computed using a highly simplified model and the force sensed from the environment, $\hat{\mathbf{n}} = \mathbf{f}_s / \|\mathbf{f}_s\|$. The calculation of $\hat{\mathbf{n}}$ is obtained through a moving average filter with a width of 30 samples obtained at a frequency of 1kHz. This model assumes negligible contact friction. During experiments, Glycerin was used as a highly lubricious medium to approximate this assumption. This is a reasonable approximation to lubricious tissue covered with bodily fluids during surgery.

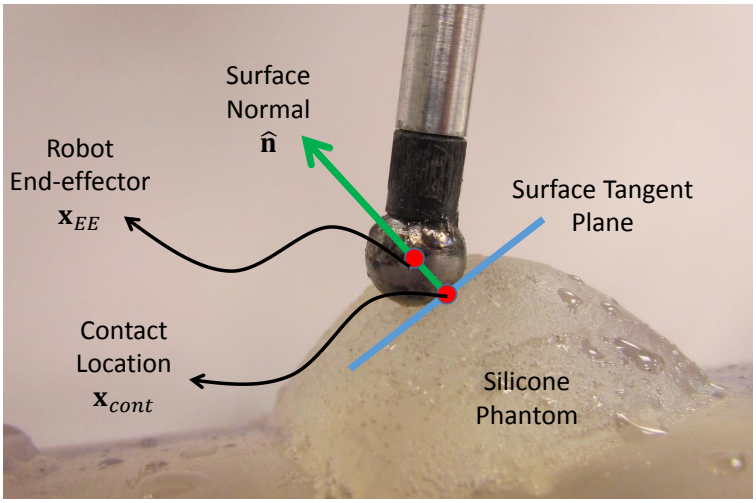


Figure 2.10: Contact Location and Surface Norm Estimation

Because friction does not dominate, this assumption does not impede task execution.

However, for a more robust/generalized formulation there are a number of options to incorporate friction compensation for better surface normal execution in task completion. A simple method is to remove force projections in the direction of the tool tip velocity [108]. Other methods include the estimation of a constraint jacobian [68], force and position sensor fusion [107], or an adaptive learning controller for surface normal estimation [45]. These and similar methods can be adjusted and extended to fit the presented task of palpation of a flexible environment with unknown geometry. However, a full exploration of this topic is beyond the scope of this chapter.

The offset of the contact location with respect to the robot end-effector can be computed as $\mathbf{x}_{cont} = \mathbf{x}_{EE} - \hat{\mathbf{n}}r$. As shown in Fig. 2.10, when the robot is in contact with environment during the exploration, it is controlled such that its motion is constrained in the surface tangential plane and its force projection onto the surface normal is regulated to a specified magnitude by *TeleOp*.

2.3.2 Wrist Orientation Optimizer for Exploration

Different from the Cartesian robot in Fig. (2.2, 2.10) where only positions can be controlled, in the case of PSM control, the wrist orientation may be optimized for exploration advantage. As shown in Fig. 2.11, the robot base frame is denoted as $\{0\}$, the gripper frame as $\{G\}$ and the exploration environment frame as $\{E\}$. The $\hat{\mathbf{z}}$ axis of frame $\{E\}$ is determined by the surface normal direction $\hat{\mathbf{n}}$ while the $\hat{\mathbf{x}}$ axis is defined as the projection of exploration moving direction onto the surface tangential plane.

We propose two criterions to optimize the wrist orientation while exploring the environment. The first criterion is to align the force probe direction (gripper frame $\hat{\mathbf{z}}_g$) with the surface normal direction $\hat{\mathbf{n}}$. This is illustrated in Fig. 2.11(a), where the angle γ is marked as the minimization goal. It is desired because it minimizes the chance of other area outside of the probing ball touching the environment, which guarantees exploration contacts to be accurately on the probe sphere surface. The second criterion is to align the highest wrist

stiffness direction with the exploration moving direction, which is illustrated as minimizing the angle ϕ in Fig. 2.11(b). The angle ϕ is defined between the axis $\hat{\mathbf{y}}_e$ and the projection of the gripper frame axis $\hat{\mathbf{x}}_g$ onto the surface tangential plane. We formulate this as a constrained optimization problem:

$$\begin{aligned} \underset{\mathbf{R}_g}{\text{maximize}} \quad & \hat{\mathbf{n}}^{\mathrm{T}}(-\hat{\mathbf{z}}_g) + \hat{\mathbf{y}}_e^{\mathrm{T}}\,\hat{\mathbf{x}}_{g_\perp} \\ \text{subject to} \quad & \dot{\mathbf{x}} = \dot{\mathbf{x}}_{des} \end{aligned} \tag{2.2}$$

where $\hat{\mathbf{x}}_{g_\perp} = \bar{\Omega}\hat{\mathbf{x}}_g$ and $\bar{\Omega}$ is defined in equation (2.1).

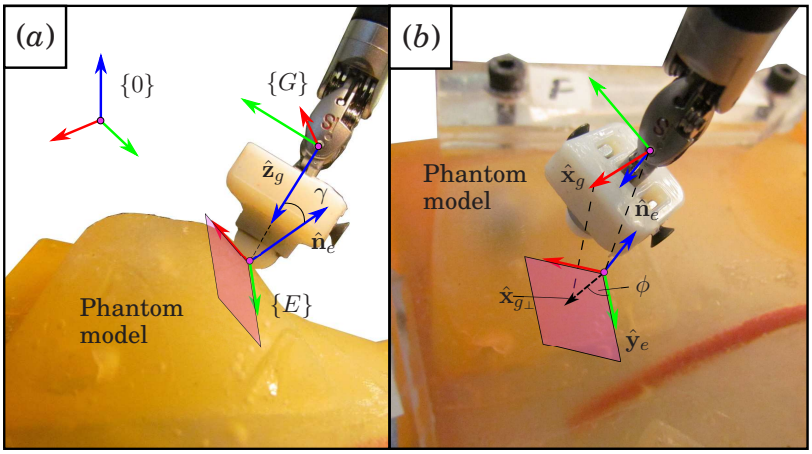


Figure 2.11: Orientation optimization for force-controlled exploration using a robot with wrist.

2.3.3 Surface Exploration

Fig. 2.12a shows the path planning to explore the entire area of interest. The path can be given in an arbitrary plane and in experiments to optimize the scan resolution we chose a plane that was parallel to XY plane in the robot base. The user selected a starting point location and several via points as the reference points, shown as a red point and several green points in Fig. 2.12a. A 2D projection onto the robot base XY plane of these reference points were used to automatically generate a raster scan pattern coordinates $\mathbf{P}_{XY} \in \mathbb{R}^{N_p \times 2}$ that enclosed several "patches" where N_p is the number of reference points, e.g. in Fig.

2.12a $N_p = 148$. And $\mathbf{P}_{XY}$ combined with the current Z coordinate of the robot in real-time were sent as command positions to the slave MLC, which implements hybrid motion/force control for the Cartesian slave robot and hybrid position/admittance control for the dVRK PSM.

The 2D scan pattern $\mathbf{P}_{XY}$ was executed at a constant velocity of 1 mm/sec in the Cartesian robot case while the exploration speed in PSM was 4 mm/sec. Currently in the Cartesian robot, without acceleration measurement feedback, a very slow execution speed was selected to avoid pseudo force disturbance to contact and surface normal estimation stemming from loading mass dynamics. In future work, an inexpensive accelerometer may be installed to provide acceleration measurement. Hence the compensation of the dynamic effect from the loading mass can be provided in real-time, enabling a much faster scanning capability. Fig. 2.12 shows the actual estimated contact locations during the force-controlled exploration using the Cartesian robot and Fig. D.2 shows the results using dVRK PSM robot. Demonstrations of force-controlled explorations using Cartesian robot and dVRK PSM are available in *Multimedia Extension II*[2] and *Multimedia Extension III*[3] respectively.

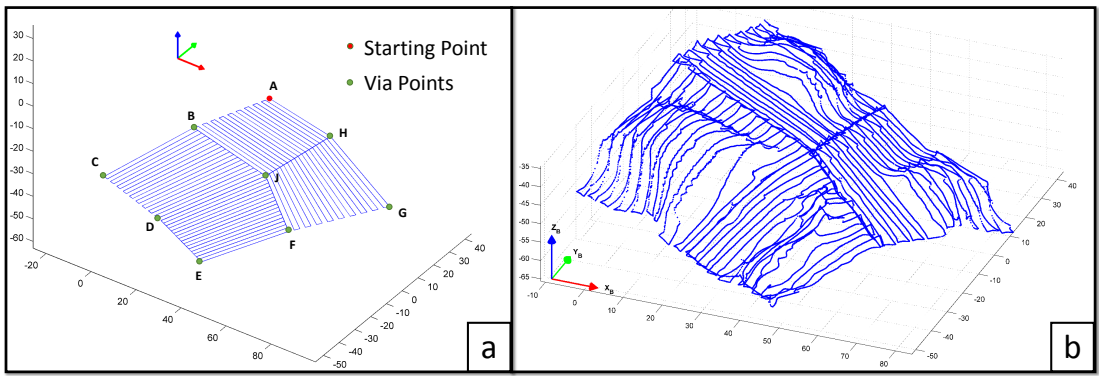


Figure 2.12: Force-controlled exploration using custom Cartesian Robot: (a) is the planned scan pattern, (b) is the actual scanned point cloud.

[2]http://arma.vuse.vanderbilt.edu/images/stories/videos/long_jmr_mme2.mp4
[3]http://arma.vuse.vanderbilt.edu/images/stories/videos/long_jmr_mme3.mp4

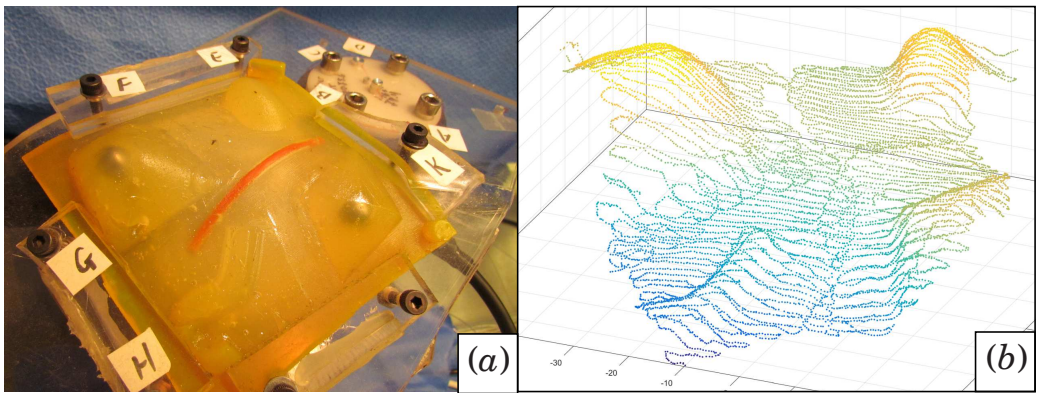


Figure 2.13: Force-controlled exploration using dVRK PSM: (a) is the silicone phantom organ, (b) is the actual collected point cloud from exploration.

2.4 Deformable Model Registration

Given the environment geometry based on force-controlled exploration data, this section presents the use of this data to update a pre-planned VF descriptor. In this work, a 3D *target curve* is used to describe a VF representing a pre-planned ablation path.

*2.4.1 Incorporating the Virtual Fixture Target Curve to an **a-priori** Model*

The following is a description of how the pre-planned VF target curve was incorporated into an *a-priori* model. An STL file representing a non-deformed silicone phantom model was obtained from a CAD model using Creo Parametric™. This STL file will henceforth be called the *a-priori* model having a corresponding point cloud (P_a). The non-deformed silicone model was laser scanned using a Faro Arm Fusion® resulting in a point cloud model (P_{ls}). A VF curve (C_{dig}) denoting a mockup pre-operative plan was also marked on the non-deformed silicone model and digitized using the Faro Arm. The laser scanned point cloud (P_{ls}) of the non-deformed silicone model and the the digitized VF curve points (C_{dig}) are shown in Fig. 2.14.

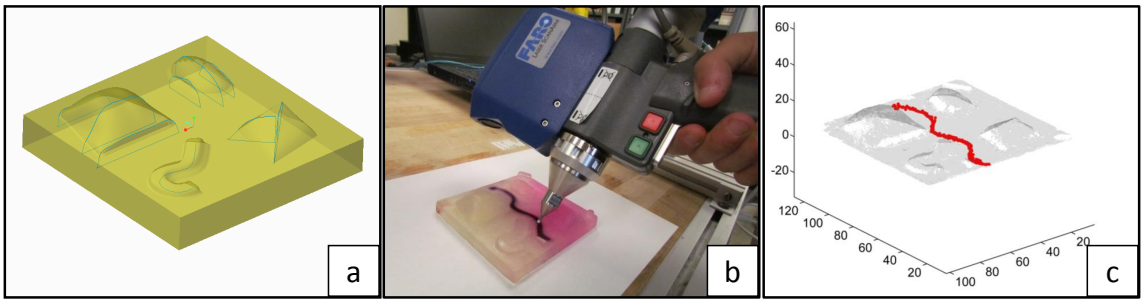


Figure 2.14: Creating an *a-priori* model of the silicone phantom
a) *a-priori* STL model (P_A); b) digitizing the target curve (C_{dig}) using Faro Arm; c) laser scan (P_{ls}) and the digitized curve (C_{dig}) in red

A deformable registration method (denoted by *DReg*) based on Coherent Point Drift [65] was used. Given two point clouds or models X_1 and X_2, this method produces a deformable registration transformation $\mathrm{T}(\cdot) \leftarrow DReg(X_1, X_2)$, such that $\mathrm{T}(X_2) \approx X_1$. Using this approach, the target curve was registered to the *a-priori* model using the laser scan (P_{ls}), by following three steps:

i) The laser scanned point cloud of the non deformed phantom (P_{ls}) was registered to the *a-priori* STL point cloud (P_a) using *DReg*. This step resulted in a transformation T_1, such that $P_a = T_1(P_{ls})$.

ii) T_1 was used to transform the digitized target curve (C_{dig}), resulting in a registered digitized target curve (C_{dig_a}) in the frame of the *a-priori* model.

iii) A polynomial basis was used to fit a smooth target curve (C_a) to (C_{dig_a}). The curve C_a is represented by a high-density of points having constant arc-length spacing along the smooth curve. The root mean square (RMS) error of target curve fitting process was recorded.

Although in i) one could have used rigid-body point-cloud registration, it was easy to use *DReg* because it also covers the special case of rigid point cloud registration while dealing with the fact that the laser point cloud contains noisy data. The final output is the *a-priori* model M_a including P_a and C_a registered in the *a-priori* model frame.

2.4.2 *Updating the Virtual Fixture Curve*

The approach to registering and updating the surgical plan as represented by the target curve (VF curve) is depicted graphically in Fig. 2.15. The deformed silicone model was explored using the robot and a point cloud $\left(\tilde{P}_a\right)$ was obtained[4]. The same deformable registration method (*DReg*) was used between the *a-priori* model point cloud (P_a) and the robot exploration data of the deformed phantom model ($\tilde{P}_a$):

$$[T_2,\ L_{\text{corr}}] \leftarrow DReg\left(\tilde{P}_a, P_a\right) \tag{2.3}$$

where T_2 is a deformable registration transformation and L_{corr} is a list of point indices relating points in P_a to their corresponding points in $\tilde{P}_a$. Applying T_2 to P_a and C_a results in the model point cloud and the VF curve points ($\breve{P}_a$) and ($\breve{C}_a$), respectively[5].

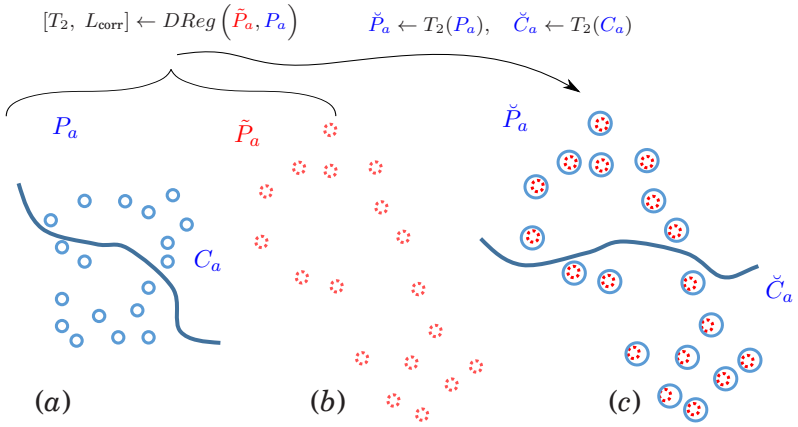


Figure 2.15: The process of updating the virtual fixture (VF) geometry: (a) the a-priori model (pre-operative model) with a VF curve, (b) the deformed environment obtained from exploration data, (c) using correspondence list to find the VF points in the exploration data set that match the curve from a-priori data set, (d) the transformed and registered VF geometry in the deformed environment

$$\breve{P}_a \leftarrow T_2(P_a), \qquad \breve{C}_a \leftarrow T_2(C_a) \tag{2.4}$$

[4]The wave accent ($\tilde{}$) denotes data obtained from the deformed phantom model using either force-controlled exploration or laser scanning

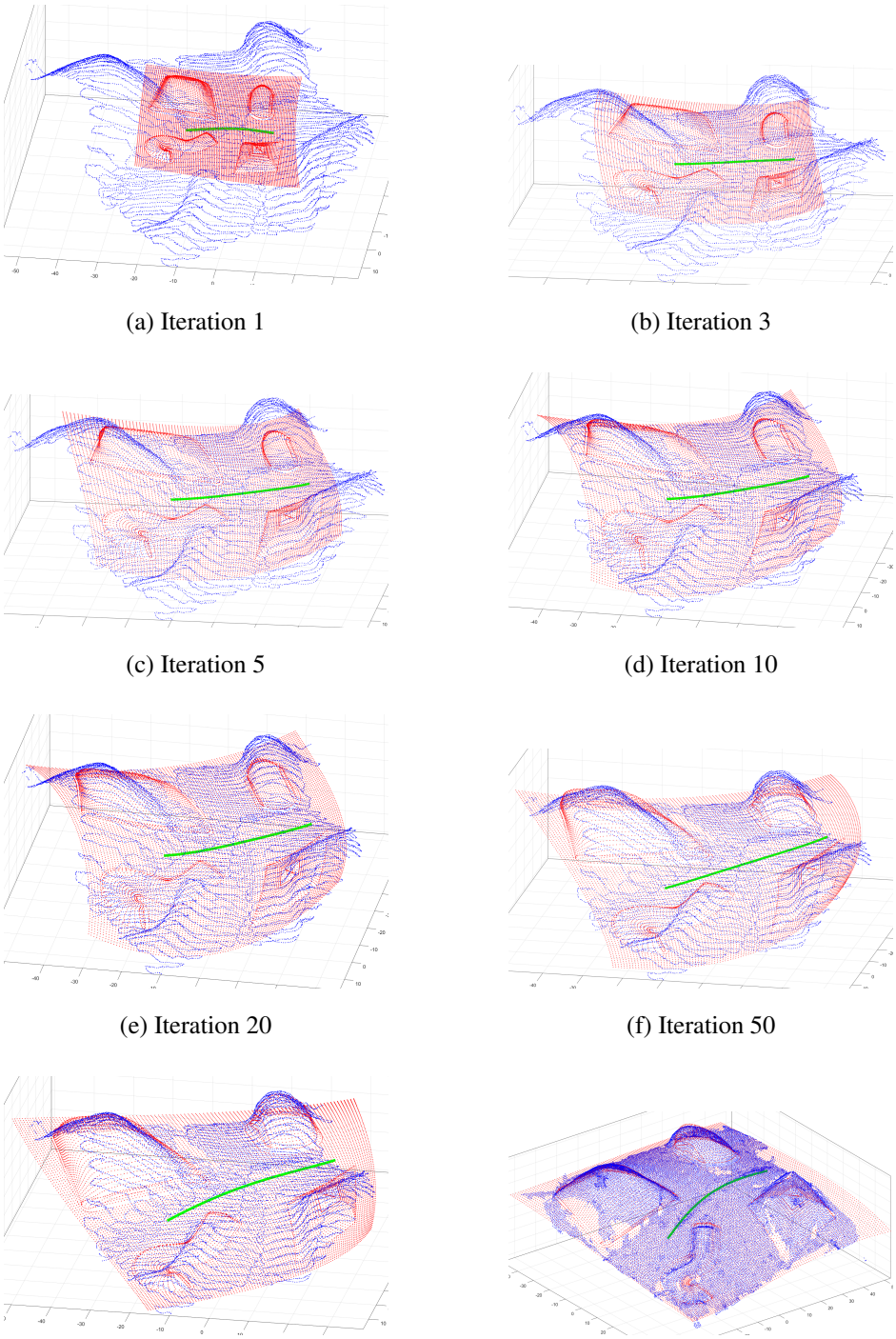
[5]The breve accent ($\breve{}$) denotes the updated *a-priori* data.

Figure 2.16 illustrates the deformable registration process using a data set from based on dVRK PSM force-controlled exploration. The blue point cloud is the exploration data, the red point cloud is the updated *a-priori* model and the green curve is the updated VF curve. A sequence of the intermediate registered results from different iterations throughout the process are presented, from which one can see the converging of the updated model to the exploration data.

To evaluate the performance of the exploration and registration, the actual VF curve on the deformed environment is digitized and a fitted ground truth curve $\tilde{C}_{gt}$ is obtained. The **ground truth VF curve fitting residual error** ε_{gt} is defined as the least-squares curve fitting residual, which is calculated as the root mean squares (RMS) error between the digitized points and their corresponding closest points on $\tilde{C}_{gt}$. The **VF curve registration error** ε_o is captured between the updated VF curve $\breve{C}_a$ and the ground truth fitted curve $\tilde{C}_{gt}$ as the following:

$$\varepsilon_o \triangleq \sqrt{\frac{1}{N}\sum_{i=1}^{N} \|\breve{c}_{a_i} - \tilde{c}^*_{gt_i}\|^2},\ \tilde{c}^*_{gt_i} = \underset{n}{\operatorname{argmin}} \|\tilde{c}_{gt_n} - \breve{c}_{a_i}\| \tag{2.5}$$

where N designates the number of sample points along the deformably registered VF curve $\breve{C}_a$ and $\tilde{c}^*_{gt_i}$ is a point along the ground truth VF curve $\tilde{C}_{gt}$ that is closest to the i^{th} sample point along the deformably registered VF curve $\breve{C}_a$.



(a) Iteration 1

(b) Iteration 3

(c) Iteration 5

(d) Iteration 10

(e) Iteration 20

(f) Iteration 50

(g) Iteration 100

(h) Iteration 100 (laser data)

Figure 2.16: Deformable registration using exploration data from dVRK PSM robot and ground truth: (a)-(g) show iterations of deformable registration using PSM robot data where iteration numbers are $\{1, 3, 5, 10, 20, 50, 100\}$, (h) is the deformable registration result using laser scan data.

2.4.3 Validation of VF Update and Discussion

The VF update strategy was validated using a custom Cartesian robot and the da Vinci research kit patient side manipulator (dVRK PSM). For both experiments we used two similar silicone phantom models made using the same mold, but with slight variations due to molding accuracy and placement/deformation of the phantom model when fixed to a deformed base as seen in Fig. D.2. It was not possible to use the same exact phantom model due to the degradation of the first silicone phantom model during the period of porting our experiments to the dVRK PSM.

The Cartesian robot was used initially because it provides both high rigidity and positional accuracy during force-controlled exploration. The force-controlled robot exploration strategy was presented in section 2.3 and the data collection can be seen in Fig. 2.12 and in *Multimedia Extension II*[6]. The same strategy was repeated on another robot - dVRK PSM, to test feasibility of our approach on a clinically relevant setup and to test whether the compliance of the dVRK PSM would prohibit the deployment of our approach for exploration-based VF update. *Multimedia Extension III*[7] shows the force-controlled exploration using the dVRK PSM.

To determine whether the force-controlled exploration affects the registration due to local deformation caused by the exploration probe, we carried out two other experiments using non-contact laser scanning. These two laser scans of each phantom model were carried out to provide comparison baselines for both of the Cartesian and PSM force-controlled exploration of the deformed phantom models. Given the laser scanned model and the force-controlled exploration data, the steps described above in equations (2.3)-(2.4) were carried out. The resulting deformed VF curve $\breve{C}_a$ based on the laser scan was then used to calculate the registration error according to equation (2.5).

Using the Faro Arm Fusion®, the ground truth VF curve $\tilde{C}_{gt}$ was digitized for phantom

[6] http://arma.vuse.vanderbilt.edu/images/stories/videos/long_jmr_mme2.mp4
[7] http://arma.vuse.vanderbilt.edu/images/stories/videos/long_jmr_mme3.mp4

Table 2.1: Deformable registration results for the Cartesian robot and dVRK PSM. For each phantom model the first column shows VF update errors based on force-controlled exploration and the second column shows the same based on laser scanning

Error Types	RMS Errors for Different Registrations [mm]			
	Phantom #1		Phantom #2	
	Cartesian	Laser	PSM	Laser
Overall ε_o	3.393	3.069	3.386	1.295
Ground Truth ε_{gt}	0.998	0.998	1.711	0.560

1. This was possible because the Cartesian robot was calibrated and its geometry allows easy registration of its base frame to the base of the Faro arm. This was not the case for phantom model 2 used with the dVRK PSM since it is an un-calibrated robot presenting substantial difficulties in registering its base frame to the base frame of the Faro arm. We therefore used the dVRK PSM as a digitizer to obtain the ground truth VF curve $\tilde{C}_{gt}$.

Table 2.1 shows the comparison of the VF curve registration errors for both the Cartesian and the dVRK PSM (also see Fig. 2.17). Since we used two similar - yet different - phantom models we had to laser scan and digitize the VF curve on each phantom model. Therefore, the table is split into left and right two-column blocks associated with each phantom model. Phantom 1 was used with the Cartesian stage robot and Phantom 2 was used with the dVRK PSM. The right and left columns of each of the two-column blocks report the VF curve registration errors using non-contact laser scan and force-controlled exploration, respectively.

The overall error ε_o for different experiments should be reviewed while considering the ground truth digitization curve fitting error ε_{gt}. For example in the PSM exploration case, ε_o has a value of 3.4 mm while ε_{gt} is 1.7 mm which contributes to the overall error. This ground truth fitting error is also consistent with the finding in [52] which shows that PSM has an approximate fiducial localization error of 1 mm. In the experiments associated with Phantom 1, the similar registration errors between the the Cartesian robot and the laser scan confirms that the effect of force-controlled exploration on the registration was negligible.

However, comparing the PSM and its Laser baseline, we find that the errors when using the PSM were significantly larger compared to when using the laser scanner. Two possible sources of error that explain this phenomenon were noticable: 1) the dVRK PSM has lateral compliance of its long and slender arms and these deflections are not observable by the robot's encoders when used as a digitizer and also when carrying out the force-controlled exploration, 2) the dVRK PSM used a wrist which had some slack due to over-use. This slack also contributes to positional error of the robot's tip. Both of robots provide overall VF update errors smaller than 3.5 mm, which is acceptable for the required registration accuracy for many abdominal surgical applications [58, 50].

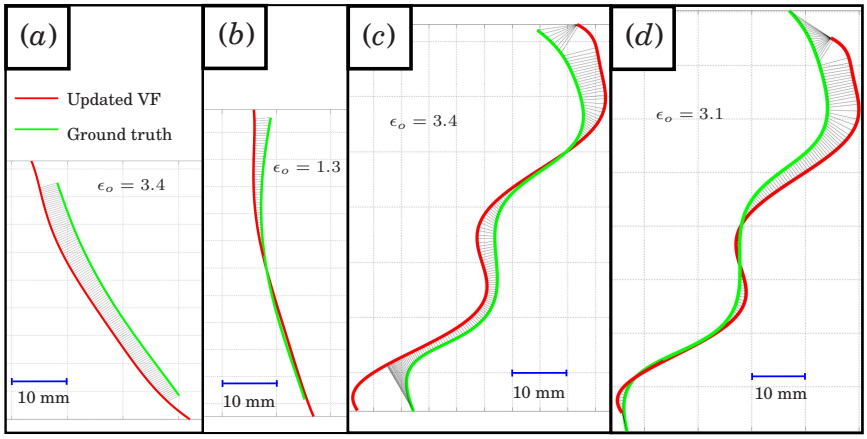


Figure 2.17: Errors between the updated VF curve and the digitized ground truth fitted curve: (a) PSM robot case, (b) PSM laser comparison, (c) Cartesian robot case, (d) Cartesian laser comparison.

2.5 Evaluation of Updated Virtual Fixtures for Mockup Ablation

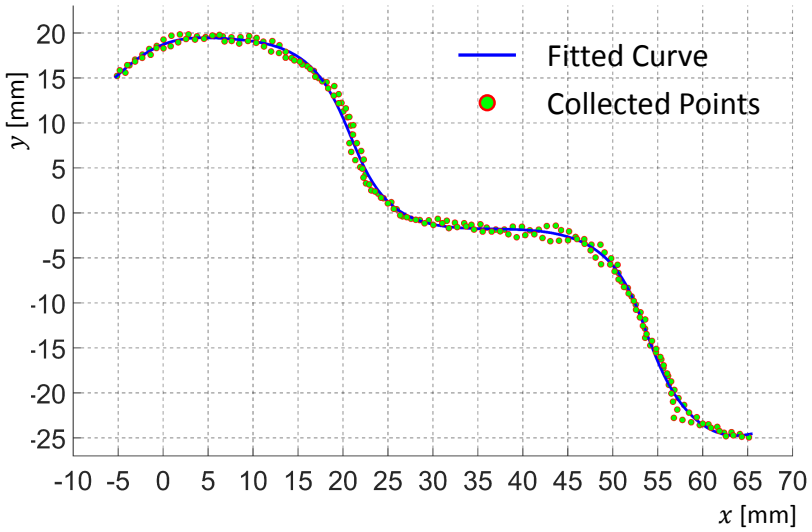


Figure 2.18: Ground truth digitization data points and the smooth curve fit

In addition to testing the feasibility of VF update using force-controlled exploration, we tested the feasibility of using such updated VF in an assistive VF in which lateral deviation from the desired VF path are resisted by applying a corrective force at the impedance master of the dVRK. The motion of the slave robot along the local surface normal is regulated by a force controller law that maintains a constant ablation force along the pre-planned ablation path.

The experimental validation was carried out using the Cartesian robot with a dVRK master. Three users participated the experiments: one user was experienced with the system and the other two were not. All users were given 30 minutes each to warm up and get used to telemanipulating the system with and without VF assistance. In both cases the hybrid force/motion controller was used with a force reference command of 0.7 N normal to the silicone phantom surface. Each user was instructed to follow a target curve back and forth twice while not paying attention to stopping exactly at the ends of the curve but rather trying to follow the curve the best they can with minimal time. Visualization was provided through an HD camera. The package *rosbag* was used to record the time and slave pose

Table 2.2: Trial completion time for each user (subject) with and without virtual fixture assistance.

	Trial Completion Time [sec]					
	Without virtual fixture			With virtual fixture		
Trial #	User 1	User 2	User 3	User 1	User 2	User 3
1	18.21	22.53	36.24	17.37	14.19	28.81
2	15.68	19.37	38.01	17.20	9.32	20.67
3	13.65	15.34	32.21	15.13	11.47	25.79
4	12.14	17.33	38.69	14.52	10.39	27.12
5	11.90	14.86	29.69	10.94	10.58	26.93
Average	14.32	17.89	34.97	15.03	11.19	25.86

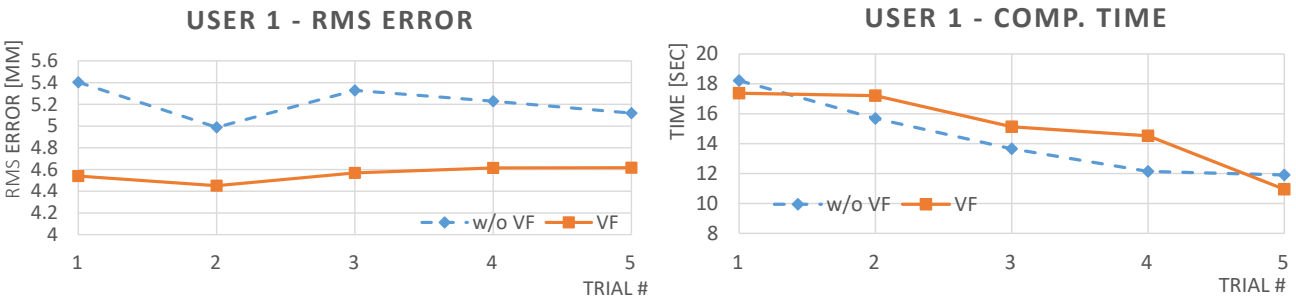
Table 2.3: RMS target curve tracking errors for each user (subject) with and without virtual fixture assistance.

	Average Trial RMS Error [mm]					
	Without virtual fixture			With virtual fixture		
Trial #	User 1	User 2	User 3	User 1	User 2	User 3
1	5.40	5.87	5.06	4.54	4.51	6.37
2	4.98	5.63	5.13	4.45	4.85	4.21
3	5.32	5.30	5.00	4.56	4.47	4.40
4	5.22	5.56	4.63	4.61	4.51	4.32
5	5.11	4.85	4.73	4.61	4.71	4.42
Average	5.21	5.44	4.91	4.55	4.61	4.75

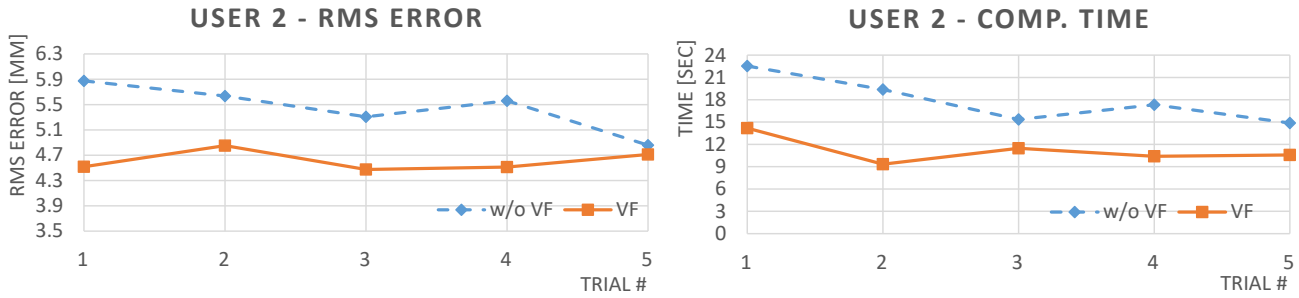
and slave command. Using the digitized data for the ground truth curve, we fitted a 10^{th} order Bernstein polynomial curve to have a smooth descriptor of the ground truth curve. This fitting process is shown in Fig. 2.18. For each user, the distance from the actual curve to the target curve was calculated after truncating the curve edges represented by the x coordinate $-5 < x < 65mm$. This effectively eliminated the edge effects since the users were not instructed to stop at the end of the curve. We also projected the data points onto a plane that best fits the target data and calculated the average RMS error along the sampled points of the curve.

The RMS tracking error and the completion time of each trial by every user is shown in Table 2.2 and Table 2.3 respectively. A comparison of each user's performance with and without VF assistance is also presented in Fig. 2.19. These results show that all

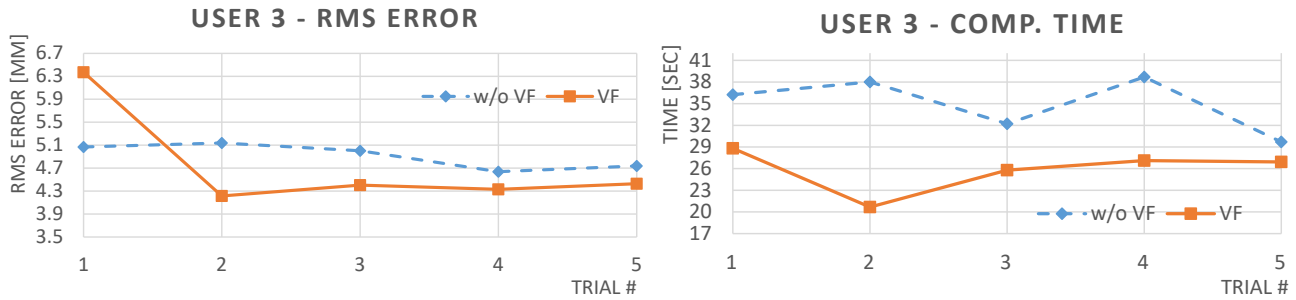
users benefited from reduced time for each trial and increased tracking accuracy when the curve-following VF law was implemented. A paired t-test between both RMS error groups resulted in a p score of 0.0031 thus rejecting the null hypothesis that both data sets come from the same distribution. Similarly, the results for time resulted with a p score of 0.0033.



(a) RMS Error and completion time of user 1.



(b) RMS Error and completion time of user 2.



(c) RMS Error and completion time of user 3.

Figure 2.19: Comparisons of the users' performance with and without virtual fixture on target curve tracing RMS errors and completion time.

2.6 Conclusions

The chapter presented a framework for updating the geometry of a virtual fixture in a deformed environment by using information from a force-controlled exploration. A modular telemanipulation framework was presented within the context of model-mediated telemanipulation. The model update employed a deformable registration based on the coherent point drift algorithm, registering an *a-priori* model of the environment and an associated virtual fixture using the exploration data. The approach was successfully demonstrated using a custom Cartesian slave robot and a da Vinci Research Kit Patient Side Manipulator which were tele-manipulated using a da Vinci Research Kit Master Tool Manipulator. The results show that the assistive behaviors after the model update benefit the users in both speed and accuracy. We believe that this framework will benefit future surgical applications where force-controlled ablation and dissection along anatomical paths is required. Future work will include using stiffness information to drive the registration method and to update the model of the environment.

2.7 Relevant Published Work

Published journal articles include:

J.1 **Wang, L.**, Chen, Z., Chalasani, P., Yasin, R. M., Kazanzides, P., Taylor, R. H. and Simaan, N., "Force-Controlled Exploration for Updating Virtual Fixture Geometry In Model-Mediated Telemanipulation", ***Journal of Mechanisms and Robotics***, Vol. 9, No. 2, pp. 021010, 2017.

J.2 Chalasani, P., **Wang, L.**, Yasin, R. M., Simaan, N., and Taylor, R.H., "Preliminary Evaluation of an Online Estimation Method for Organ Geometry and Tissue Stiffness", ***IEEE Robotics and Automation Letters***, vol. 3, no. 3, pp. 1816-1823, July 2018.

J.3 Ayvali, E., Ansari, A., **Wang, L.**, Simaan, N., Choset, H., "Utility-Guided Palpation for Locating Tissue Abnormalities," ***IEEE Robotics and Automation Letters***, Vol. 2, No. 2, pp. 864-871, 2017.

J.4 Simaan, N., Yasin, R. M. and **Wang, L.** , "Medical Technologies and Challenges of Robot-Assisted Minimally Invasive Intervention and Diagnostics", ***Annual Review***

of Control, Robotics, and Autonomous Systems 1 (2018): 465-490.

Published conference papers include:

C.1 **Wang, L.**, Chen, Z., Chalasani, P., Pile, J., Kazanzides, P., Taylor, R.H., Simaan, N., "Updating Virtual Fixtures From Exploration Data in Force-Controlled Model-Based Telemanipulation", Proceedings of ASME International Design Engineering Technical Conference (IDETC'2016), Charlotte, NC., 2016.

C.2 Yasin, R. M., **Wang, L.**, Abah, C. and Simaan, N., "Using Continuum Robots for Force-controlled Semi-autonomous Organ Exploration and Registration", IEEE International Symposium on Medical Robotics (ISMR), Atlanta, GA, 2018.

C.3 Chalasani, P., **Wang, L.**, Roy, R., Simaan, N., Taylor, R.H., Kobilarov, M., "Concurrent Nonparametric Estimation of Organ Geometry and Tissue Stiffness Using Continuous Adaptive Palpation", IEEE International Conference on Robotics & Automation (ICRA'), Stockholm, Sweden, 2016.

C.4 Srivatsan, R.A., **Wang, L.**, Ayvali1, E., Simaan, N., Choset, H., "Simultaneous Registration and Stiffness mapping of a Flexible Environment using Stiffness and Geometric Prior," Proceedings of the Hamlyn Symposium of Medical Robotics, London, U.K., 2016.

C.5 Srivatsan, A., Ayvali, E., **Wang, L.**, Roy, R., Simaan, N., Choset, H., "Complementary Model Update: A Method for Simultaneous Registration and Stiffness Mapping in Flexible Environments" IEEE International Conference on Robotics & Automation (ICRA'), Stockholm, Sweden, 2016.

C.6 Ayvali, E., Srivatsan, A., **Wang, L.**, Roy, R., Simaan, N., Choset, H., "Using Bayesian Optimization to Guide Probing of a Flexible Environment for Simultaneous Registration and Stiffness Mapping", IEEE International Conference on Robotics & Automation (ICRA'), Stockholm, Sweden, 2016.

C.7 Salman, H., Ayvali, E., Srivatsan, R. A., Ma, Y., Zevallos, N., Yasin, R.M., **Wang, L.**, Simaan, N., and Choset, H., "Trajectory-Optimized Sensing for Active Search of Tissue Abnormalities in Robotic Surgery", IEEE International Conference on Robotics & Automation (ICRA'), 2018.

Chapter 3

Geometric Calibration of Continuum Robots

3.1 Related Work

The vast majority of works assume that the segments of continuum robots (CRs) bend in constant curvature, [98]. This assumption can work if certain design conditions hold: e.g. very close spacing between spacer disks of an MBCR and negligible gravitational effects [102]. These assumptions limit the design space of continuum robots in favor of computational and modeling simplicity. In practical implementation of such robots, stress concentrations, internal friction and imperfect assembly can cause the non-circular bending; therefore, introducing modeling errors and limiting the accuracy of these robots.

When calibrating continuum robots, there are two mappings to rectify: (i) configuration space to task space, (ii) configuration space to joint space. Figure 3.1 shows an illustration of the *configuration space* of a two-segment MBCR. The configuration space refers to geometric variables characterizing the shape of the continuum segment (for example bending angle θ_e and the angle of the plane in which the segment bends δ_0 [88]). The configuration space to task space mapping may have errors due to non-circular bending, coupling effects between segments, and assembly errors (e.g. a snake segment that is assembled with a slight twist along its backbone). The configuration to joint space mapping may have errors due to geometric and joint home position uncertainties. For the shape modeling of hyper-redundant robots, works such as [17, 18] suggested a modal representation while others like [111] used a spline method. These approaches may be adapted to capture the kinematics of continuum robots (e.g. [112, 33]). Also, recent works extended beyond the constant curvature assumption by approximating the shape of a single continuum segment as a set of constant curvature sub-segments (e.g. [51, 62]). Exact modeling approaches using elliptic integrals for statics of MBCRs [103] and Cosserat rod theory for dynamics of wire-actuated SBCRs [78, 75] have also been presented. These modeling approaches however cannot

capture modeling errors due to joint homing uncertainty or twisted assembly.

Works on geometric calibration of CRs are sparse and design-specific. Webster [99] calibrated the shape and torsion of concentric tubes using constant curvature assumption. Camarillo [12] used vision to calibrate coupling effects in wire-actuated two-segment catheters. Escande [27] formulated a calibration framework that also assumes constant curvature for each continuum segment of an SBCR. Murphy [64] characterized the kinematics of a flexure-based SBCR using image analysis and constrained optimization. Jang [41] presented a generalized Jacobian for concentric tubes toward on-line parameter estimation using an extended Kalman filter. Our previous work [95] where a parameterized modal approach similar to [18] was used to formulate the kinematics and to investigate the error prorogation in simulation base. This work was limited to considering only the derivation of the configuration-to-joint space mapping for twisted configuration of these robots. This work builds on and extends our prior work to include generalized treatment of geometric calibration of MBCR's.

3.2 Problem Statement, Nomenclature, and Assumptions

Figure 3.1 shows an example of a two segment MBCR. Each segment is comprised of a central backbone that is surrounded by equidistant secondary backbones. All backbones are generally made of superelastic NiTi and pass through as series of spacer disks or an elastomeric sheath that maintains the radial distance between the backbones. The backbones terminate and connect to an end disk. By actuating the secondary backbones, an MBCR segment can be controllably bent in two degrees-of-freedom (DoF). In some recent embodiments a design variation was presented to allow backbone extensibility (e.g. [69]), or unconstrained backbones (e.g. [71]).

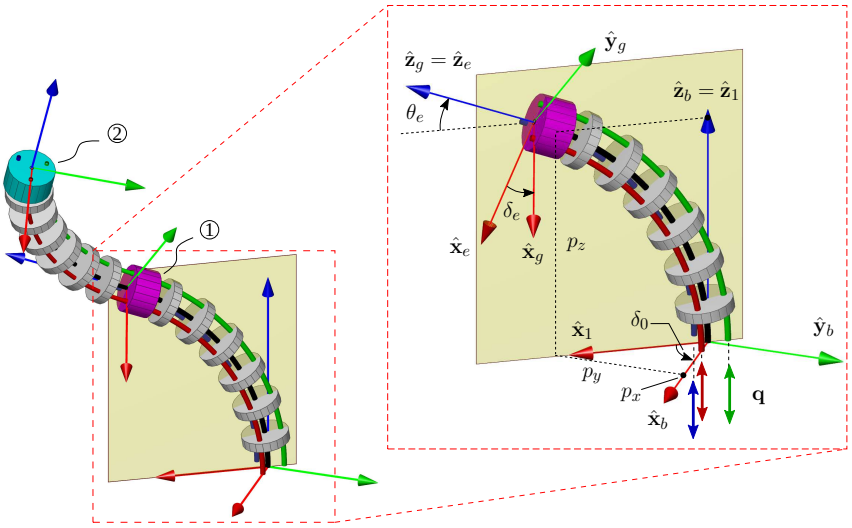


Figure 3.1: An example of a 2-segment MBCR:
1 - End disk of 1st segment; 2 - End disk of 2nd segment

3.2.1 Problem Statement

The problem of manipulator calibration is accomplished by solving three sub-problems: the parameterization of a kinematic model to include the calibration unknowns, the error propagation from parameters to measurable variables, and an estimation approach to find the calibration parameters.

The *parameterization problem* is: given an n-segment MBCR with $\mathbf{x}$ and $\mathbf{R}$ designating the position and orientation of its end effector and $\boldsymbol{\psi}_i = [\theta_{e_i}, \delta_i]^{\mathrm{T}}, i = 1 \ldots n$ designating its configuration vector, find the expressions of the *configuration* to *task* space mapping $\mathcal{F}_{\mathrm{T}}$ and the *configuration* to *joint* space mapping $\mathcal{F}_{\mathrm{q}}$ in a way that accounts for bending shape deviation, joint space errors and twisting:

$$\begin{cases} \boldsymbol{\mathcal{F}}_{\mathrm{q}} : (\boldsymbol{\psi}, \mathbf{k}) \rightarrowtail \mathbf{q} & \text{config. to joint space} \quad (3.1a) \\ \boldsymbol{\mathcal{F}}_{\mathrm{T}} : (\boldsymbol{\psi}, \mathbf{k}) \rightarrowtail (\mathbf{x}, \mathbf{R}) & \text{config. to task space} \quad (3.1b) \end{cases}$$

where the joint space vector $\mathbf{q} \triangleq [q_1, q_2, q_3]^{\mathrm{T}}$ represents the displacements of push/pull on the three secondary backbones, as defined in (3.52), and $\mathbf{k}$ represents the geometric calibration parameters, as introduced in (3.53).

The *error propagation problem is:* given the two mappings $\mathcal{F}_{\mathrm{q}}$ and $\mathcal{F}_{\mathrm{T}}$, investigate the error propagation in these mappings as a function of small changes in the vector of calibration parameters $\mathbf{k}$. This results in *identification Jacobians* mapping a differential change in calibration parameter space to a differential in kinematic variable space.

The *Calibration problem statement is:* given a set of identification Jacobians along with measurements of kinematic variables (e.g. measurements of joint positions and end effector positions/orientations provided by a motion sensor attached to the robot) find the best estimate of the calibration parameters $\mathbf{k}$ that would best explain the deviation between the uncalibrated model and the experimental data set in a least-squares sense.

We limit the scope of this work to consider calibration of motion subject to very small or no loads (i.e. calibrating the statics/dynamics is excluded). Though this is a restrictive assumption, there are several surgical applications of MBCR's that involve no forceful interaction with tissue. These applications include minimally invasive laser ablation/resection, minimally invasive surveillance of bodily cavities and minimally invasive injection of therapeutic agents (e.g. transnasal vocal fold injection mediatization). Furthermore, the calibration framework presented herein can be used with a static model to calibrate a generalized kinematics and statics model of MBCR's subject to loading. The key limitation is that the loading should not cause a yielding failure of the MBCR structure so as to violate the geometric design constraints.

3.2.2 Kinematic Nomenclature

Figure 3.1 shows the key notation used in this chapter. Several frames are used to facilitate the kinematic derivation. Frame $\{\mathrm{B}\}$ is the base disk frame[1] having its z axis perpendicular to the base disk and its x axis pointing from the central backbone to the first backbone. Frame $\{1\}$ has its origin and $\mathbf{z}$ axis in common with frame $\{\mathrm{B}\}$ and its $\mathbf{x}$ axis pointing along the projection of axis $\mathbf{z}_g$ onto the x-y plane of frame $\{\mathrm{B}\}$, the axis $\mathbf{z}_g$ being

[1] We use the notation $\{\mathrm{A}\}$ to refer to a right-handed frame with $\hat{\mathbf{x}}_a, \hat{\mathbf{y}}_a, \hat{\mathbf{z}}_a$ as its unit vector axes and $\mathbf{a}$ as its origin

the normal to the end disk[2]. This frame is used to characterize the plane in which the snake segment bends (henceforth called *bending plane*).

We also define the end-disk frame $\{\mathrm{E}\}$ with its origin at the center of the end disk and its z axis perpendicular to the end disk as shown in Figure 3.1. The axis $\hat{\mathbf{x}}_e$ is defined as the intersection of the plane defined by the top surface of the end disk and the bending plane. Finally, we define the end effector frame $\{\mathrm{G}\}$ to share its origin and z axis with frame $\{\mathrm{E}\}$ and its x axis $\hat{\mathbf{x}}_g$ pointing from its origin to the first backbone. Frames $\{\mathrm{Es}\}$ and $\{\mathrm{Gs}\}$ are defined in similar manner as $\{\mathrm{E}\}$ and $\{\mathrm{G}\}$, but at an arc-length s that is measured along the central backbone from the base disk.

The configuration of a continuum segment is designated by the vector $\boldsymbol{\psi}$. The vector $\boldsymbol{\psi}$ includes the angles δ_0 and θ_e. The angle δ_0 is defined according to the right hand rule about $\hat{\mathbf{z}}_b$ from $\hat{\mathbf{x}}_1$ to $\hat{\mathbf{x}}_b$. The bending angle θ_e is given according to the right hand rule about $-\hat{\mathbf{y}}_1$ from $\hat{\mathbf{x}}_1$ to $\hat{\mathbf{z}}_g$. For simplicity of derivation, we later replace the use of θ_e with a parameter t that parameterizes θ_e. The angle δ_e is defined according to the right hand rule about $\hat{\mathbf{z}}_e$ from $\hat{\mathbf{x}}_e$ to $\hat{\mathbf{x}}_g$.

The joint space vector for a given segment is designated by $\mathbf{q}$ where positive values designate extension (pushing) of the secondary backbones.

The above nomenclature ignores the bending shape and the twist about the backbone of the CR segment. The following sections will discuss these effects.

3.2.3 Shape Interpolation Assumption

When a CR segment is bent, its equilibrium shape is given as a solution to the minimal energy problem. In [88] it was shown that the bending shape of a CR segment does not affect the joint space mapping $\mathcal{F}_{\mathrm{q}}$. Therefore, relying on joint and configuration space measurements of $\mathbf{q}$ and $\boldsymbol{\psi}$ is not sufficient for calibration. Since the task space mapping $\mathcal{F}_{\mathrm{T}}$ is affected by the bending shape, a parameterizations of the bending shape is needed for

[2] If $\hat{\mathbf{z}}_g$ is perpendicular to the $x-y$ plane of $\{\mathrm{B}\}$ then $\hat{\mathbf{x}}_1$ is not defined. In this case, the segment is straight and $\hat{\mathbf{x}}_1$ is not needed

Table 3.1: Nomenclature for kinematics

Symbol	Description
Frame $\{\mathrm{F}\}$	designate a right-handed frame with unit vectors $\hat{\mathbf{x}}_f, \hat{\mathbf{y}}_f, \hat{\mathbf{z}}_f$ and $\mathbf{f}$ as its origin.
Frame $\{\mathrm{B}\}$	the base disk frame with $\mathbf{b}$ located at the center of the base disk, $\hat{\mathbf{x}}_b$ passing through the first SB and $\hat{\mathbf{z}}_b$ perpendicular to the base disk.
Frame $\{1\}$	characterizes the plane in which the snake segment bends and it is obtained by a rotation of $(-\delta)$ about $\hat{\mathbf{z}}_b$. Unit vector $\hat{\mathbf{x}}_1$ is along the projection of the PB on the plane of the base disk and $\hat{\mathbf{z}}_1 = \hat{\mathbf{z}}_b$.
Frames $\{\mathrm{E}\}$ & $\{\mathrm{G}\}$	Frame $\{\mathrm{E}\}$ is defined with $\hat{\mathbf{z}}_e$ as the normal to the end disk and $\hat{\mathbf{x}}_e$ is the intersection of the bending plane and the end disk top surface. Frame $\{\mathrm{G}\}$ is obtained by a rotation angle δ_e about $\hat{\mathbf{z}}_e$ which is the unit vector normal to the end disk. This angle is given by $\delta_e = \delta_0 + \gamma_e$.
Frames $\{\mathrm{Es}\}(\mathrm{s})$ & $\{\mathrm{Gs}\}(\mathrm{s})$	associated with arc length s. These frames are defined in a manner similar to the definition of frames $\{\mathrm{E}\}$ and $\{\mathrm{G}\}$ but for a specific value of s as opposed to $s = L$. The origin of $\{\mathrm{Es}\}$ is located at point $\mathbf{x}(s)$ and obtained by integration along the PB.
$\mathbf{v}(s)$ & $\mathbf{w}(s)$	a vector $\mathbf{v}(s) = r\hat{\mathbf{x}}_{gs}$ which points from $\mathbf{x}(s)$ on the PB to the corresponding point on the first SB. The location of the point on the first SB that corresponds to point $\mathbf{x}(s)$ on the PB is designated by $\mathbf{w}(s)$. We note that $\mathbf{w}(s) = \mathbf{x}(s) + \mathbf{v}(s)$.

enabling bending shape calibration.

We assume that the equilibrium shape is configuration-dependent and that this dependence may be captured using a family of curvature profiles $\kappa(s,t)$. The arc-length parameter s is measured along the central backbone from the base disk and L is used to denote the segment length. The interpolation parameter $t \in [0,1]$ selects a specific bending shape that is interpolated from two curvature profiles (shape generators) $\kappa_a(s)$ and $\kappa_b(s)$ describing the limits of the bending workspace of the continuum segment:

$$\kappa(s,t) = t\,\kappa_b(s) + (1-t)\,\kappa_a(s) \quad t \in [0,1], s \in [0,L] \tag{3.2}$$

Referring to Fig. 3.2(a), the relationship between t and θ_e is:

$$\theta_e \;\; = \theta_0 + \textstyle\int_0^L \kappa(s,t)\mathrm{ds} \;\; = \theta_{e_a} + t\,(\theta_{e_b} - \theta_{e_a}) \tag{3.3}$$

where θ_0 is the tangent angle at $s = 0$ (in Fig. 3.2(a) $\theta_0 = \frac{\pi}{2}$); θ_{e_a} and θ_{e_b} are the tip bending angles of the shape generators.

Henceforth, we will use a re-parametrization of the configuration space variables based on Eq. (3.3) whereby the configuration space vector ψ is redefined as:

$$\psi \triangleq [t, \delta_0]^{\mathrm{T}} \tag{3.4}$$

The motivation for this re-parametrization lies in the linear relationship between the bending shape $\kappa(s,t)$ and the curve generator shapes, which is advantageous for calibration.

The shape generator functions $\kappa_a(s)$ and $\kappa_b(s)$ are described using a modal representation inspired by [18]:

$$\kappa_a(s) = \mathbf{a}^T \boldsymbol{\eta},\ \kappa_b(s) = \mathbf{b}^T \boldsymbol{\eta},\ \boldsymbol{\eta}(s) = [s^0, s^1, \ldots, s^m]^T \tag{3.5}$$

where $\mathbf{a}$ and $\mathbf{b}$ are vectors of modal factors and $\boldsymbol{\eta}(s)$ is a vector of modal functions[3]. Although the approach presented below holds for $m > 1$, we use $m = 1$ because our device exhibits a small deviation from circular shape [103].

3.2.4 Twist Assumption

Twisting about the primary backbone (henceforth referred to as *backbone twist*) of MBCRs is typically caused by inexact assembly of the spacer disks on the primary backbone or by a wrong definition of the "home" position of the joints controlling the length of the secondary backbones. For example, imagine that the secondary backbones are assembled such that they are almost equal in length but longer than the primary backbone; the continuum robot segment will start from an almost straight configuration but with a helical twist of the secondary backbones about the primary backbone.

[3]The Vandermonde matrtix corresponding with the monomial basis is not numerically stable for $m > 6$ [2]. Other better-conditioned modal bases could be used if higher orders are needed (e.g. Chebyshev polynomials)

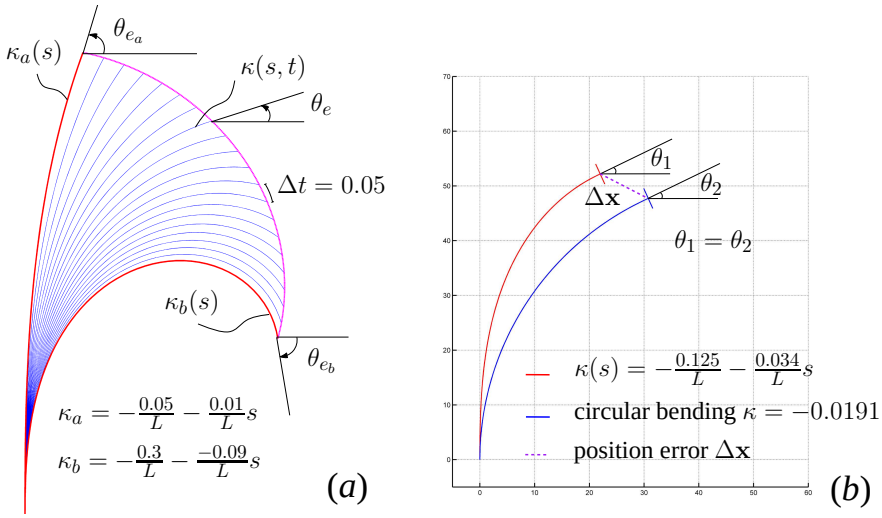


Figure 3.2: Illustration of shape interpolation (a) and shape deviation from circular bending (b).

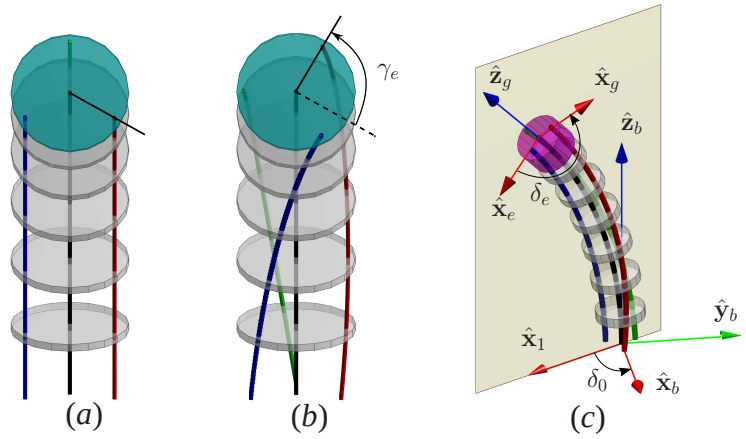


Figure 3.3: A continuum segment (a) without twist, (b) twisted by angle γ_e, (c) the definition of the configuration variables δ_0 and θ_e for a continuum segment twisted by γ_e

It is next assumed that a CR segment has a configuration-dependent twist angle $\gamma_e(\psi)$, Fig. 3.3(a-b). The choice of configuration-dependent assumption comes from an equilibrium and energy minimization point of view: a twisted snake segment experiences more twisting when it is bent due to the larger loading forces on the backbones. This configuration-dependence is captured by an assembly error offset γ_{e0} at the base disk and two linear constants k_θ and k_δ:

$$\gamma_e = \gamma_{e_0} + k_\theta t + k_\delta \delta_0 \tag{3.6}$$

Fig. 3.3(c) illustrates this twist effect where $\delta_e|_{s=0} = \delta_0$ and $\delta_e|_{s=L} = (\delta_0 + \gamma_e)$. Assuming a uniform torsional stiffness, the twist angle is linearly distributed along the arc length s:

$$\delta(s) \triangleq \delta_0 + \gamma(s), \quad \gamma(s) = \gamma_e \tfrac{s}{L} \tag{3.7}$$

where $\delta(s)$ is tantamount to δ_0 but defined for frame $\{\mathrm{Gs}\}(\mathrm{s})$. This frame describes the pose of an imaginary end-effector at arc length s with $\hat{\mathbf{z}}_{g_s}$ along the central backbone tangent and $\hat{\mathbf{x}}_{g_s}$ pointing to the first secondary backbone, Fig. 3.4.

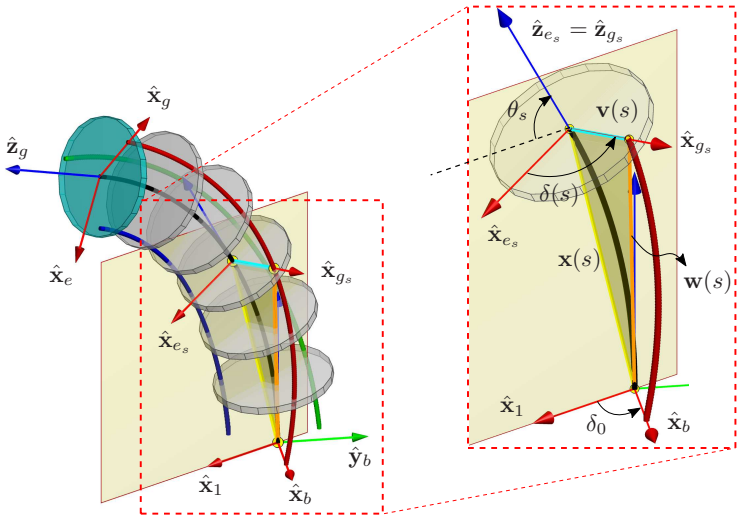


Figure 3.4: Nomenclature illustrations for a twisted snake segment: (left) shows a twisted continuum robot segment and (right) shows the vector loop to derive the secondary backbone length calculation in section 3.4.

3.3 Kinematic Error Propagation from Configuration to Task Space

We decompose the problem of formulating the *configuration*-to-*task*-space mapping $\mathcal{F}_{\mathrm{T}}$ into two sub-tasks: the in-plane bending kinematics and the end effector orientation with backbone twist. This formulation separates the calibration parameters into two groups: group 1 characterizes the bending shape and it is represented by a *shape characteristic* vector $\mathbf{k}_\kappa$. Group 2 characterizes the twist behavior and represented by a *twist characteristic* vector $\mathbf{k}_\gamma$. These vectors are defined as:

$$\mathbf{k}_\kappa \triangleq [l, \mathbf{a}^{\mathrm{T}}, \mathbf{b}^{\mathrm{T}}]^{\mathrm{T}} \tag{3.8}$$

$$\mathbf{k}_\gamma \triangleq [\gamma_{e0}, k_\theta, k_\delta]^{\mathrm{T}} \tag{3.9}$$

where l is the arc length where a motion sensor is attached, $\mathbf{a}, \mathbf{b}$ are the modal factors in (3.5) and $\mathbf{k}_\gamma$ is defined in (3.6).

This section presents the kinematics and error propagation models leading to kinematic identification Jacobians. The goal is to obtain the differential end-effector position and orientation errors due to perturbations of $\delta\mathbf{k}_\kappa$ and $\delta\mathbf{k}_\gamma$.

3.3.1 Single Segment In-Plane Kinematics with Backbone Twist

Since the CR segment is contained in the bending plane $\{\hat{\mathbf{x}}_1 - \hat{\mathbf{z}}_1\}$, we first derive its in-plane bending kinematics. The end effector position and orientation are given by:

$${}^1\mathbf{x}(s) = \int_0^s [c_{\theta_s(\tau)}, 0, s_{\theta_s(\tau)}]^{\mathrm{T}}\, \mathrm{d}\tau \qquad s \in [0, L] \tag{3.10}$$

$$\theta_s(s) = \frac{\pi}{2} + \int_0^s \kappa(\tau, t)\mathrm{d}\tau \qquad s \in [0, L] \tag{3.11}$$

where ${}^1\mathbf{x}$ denotes the coordinates of the end-effector in frame $\{1\}$[4] and $c_{(\cdot)}$ and $s_{(\cdot)}$ denote the cosine and sine functions.

[4] we use the notation ${}^a\mathbf{x}$ to designate a vector $\mathbf{x}$ described in frame $\{\mathrm{A}\}$

The end effector orientation completes the mapping $\mathcal{F}_T$. Referring to Fig. 3.4, the orientation of $\{\mathrm{Gs}\}(\mathrm{s})$ is:

$$ {}^{b}\mathbf{R}_{gs}(s) = e^{-\delta_0 [\mathbf{e}_3{}^{\wedge}]}\, e^{(\frac{\pi}{2}-\theta_s)[\mathbf{e}_2{}^{\wedge}]}\, e^{\delta(s)[\mathbf{e}_3{}^{\wedge}]} \qquad s \in [0,L] \tag{3.12} $$

where $[\mathbf{a}^{\wedge}]$ represents a skew symmetric matrix generated from vector $\mathbf{a}$, and $\mathbf{e}_i$ denotes the standard basis vectors for $\mathbb{R}^3$ (e.g. $\mathbf{e}_2 = [0,1,0]^{\mathrm{T}}$). These equations indicate that the orientation ${}^{b}\mathbf{R}_{g_s}$ can be expressed as a function of $\{\delta_0, \theta_s, \gamma_{(s)}\}$.

To facilitate the derivation of kinematic identification Jacobians in the following sections, we define $\mathbf{x}_p$ as the combined in-plane position and tip angle for a sensor located at an arbitrary location $l \in [0,L]$:

$$ \mathbf{x}_p \triangleq \left[\begin{array}{c} {}^{1}x(l), {}^{1}z(l), \theta_e(l) \end{array} \right]^{\mathrm{T}} \tag{3.13} $$

where ${}^{1}x$, and ${}^{1}z$ designate the $\mathbf{x}$ and $\mathbf{z}$ coordinates of the modeled sensor position, and the subscript $_p$ indicates "in-plane". In the framework defined in later sections, $\mathbf{x}_p$ is measured for calibration use.

Since $\mathbf{x}_p$ is unaffected by $\mathbf{k}_\gamma$ (i.e. it is unaffected by twist about the central backbone), it may be expressed in a functional form as:

$$ \mathbf{x}_p = \mathbf{f}_p(t, \mathbf{k}_\kappa) \tag{3.14} $$

Finally, to facilitate the kinematic simulations, the instantaneous kinematics (geometric) Jacobian is defined as:

$$ \mathbf{J}_{\mathbf{x}t} \triangleq \frac{\partial \mathbf{f}_p}{\partial t} = \left[\frac{\partial {}^{1}x(l)}{\partial t}, \frac{\partial {}^{1}z(l)}{\partial t}, \frac{\partial \theta_s|_{(s=l)}}{\partial t} \right]^{\mathrm{T}} \tag{3.15} $$

where details of its derivation are in Appendix B.1.

We next define the vector of *pose* as a six-dimensional parametrization of position and orientation $\boldsymbol{\xi} \triangleq [\mathbf{x}^{\mathrm{T}}, \boldsymbol{\mu}^{\mathrm{T}}]^{\mathrm{T}}$ where $\mathbf{x}$ and $\boldsymbol{\mu}$ are the position and orientation vectors (e.g. $\boldsymbol{\mu}$ can be the axis-angle vector parametrizing the orientation). The following sections derive the pose error propagation for a multi-segment continuum robot. The effects of $\delta\mathbf{k}_\kappa$ and $\delta\mathbf{k}_\gamma$ are combined through superposition with a first-order error propagation model. Defining $\mathbf{k}_T$ and $\delta\mathbf{k}_T$ for one segment as:

$$\mathbf{k}_T \triangleq [\mathbf{k}_\kappa{}^{\mathrm{T}}, \mathbf{k}_\gamma{}^{\mathrm{T}}]^{\mathrm{T}}, \quad \delta\mathbf{k}_T \triangleq [\delta\mathbf{k}_\kappa{}^{\mathrm{T}}, \delta\mathbf{k}_\gamma{}^{\mathrm{T}}]^{\mathrm{T}} \tag{3.16}$$

This allows the derivation of a combined pose error model:

$${}^b\delta\boldsymbol{\xi}(\delta\mathbf{k}_T) = {}^b\delta\boldsymbol{\xi}(\delta\mathbf{k}_\kappa) + {}^b\delta\boldsymbol{\xi}(\delta\mathbf{k}_\gamma) \tag{3.17}$$

3.3.2 Pose Error due to Shape Deviation ($\delta\xi(\delta\mathbf{k}_\kappa)$)

An *identification* Jacobian, $\mathbf{J}_{\mathbf{k}_\kappa}$ is derived to propagate $\delta\mathbf{k}_\kappa$ to the small perturbation of $\delta\mathbf{x}_p$:

$$\delta\mathbf{x}_p = \mathbf{J}_{\mathbf{k}_\kappa}\delta\mathbf{k}_\kappa, \quad \mathbf{J}_{\mathbf{k}_\kappa} \in \mathbb{R}^{3\times 5} \tag{3.18}$$

$$\mathbf{J}_{\mathbf{k}_\kappa} \triangleq \frac{\partial \mathbf{f}_p(t, \mathbf{k}_\kappa)}{\partial \mathbf{k}_\kappa} = \left[\frac{\partial \mathbf{f}_p}{\partial l}, \frac{\partial \mathbf{f}_p}{\partial \mathbf{a}}, \frac{\partial \mathbf{f}_p}{\partial \mathbf{b}}\right] \tag{3.19}$$

where the expressions for $\frac{\partial \mathbf{f}_p}{\partial l}, \frac{\partial \mathbf{f}_p}{\partial \mathbf{a}}, \frac{\partial \mathbf{f}_p}{\partial \mathbf{b}}$ may be obtained by differentiating Eq. (3.14) as detailed in Appendix B.2. For a single-segment continuum robot, the differential pose change of the end effector frame due to $\delta\mathbf{k}_\kappa$ is:

$${}^b\delta\boldsymbol{\xi}(\delta\mathbf{k}_\kappa) \triangleq \left[({}^b\delta\mathbf{x}_\kappa)^{\mathrm{T}}, ({}^b\delta\boldsymbol{\mu}_\kappa)^{\mathrm{T}}\right]^{\mathrm{T}}, \quad {}^b\delta\boldsymbol{\xi} \in \mathbb{R}^{6\times 1} \tag{3.20}$$

where subscript κ indicates that the source of error is $\delta\mathbf{k}_\kappa$.

Referring back to the definition of $\mathbf{x}_p$ in Eq. (3.13), we note that only the end effector

coordinates along $\hat{\mathbf{x}}_1$ and $\hat{\mathbf{z}}_1$ are affected by $\delta\mathbf{k}_\kappa$. Therefore, ${}^b\delta\mathbf{x}_\kappa$ may be expressed as:

$$ {}^b\delta\mathbf{x}_\kappa = {}^b\mathbf{R}_1\,\mathbf{S}_\mathbf{x}\,\delta\mathbf{x}_p, \quad \mathbf{S}_\mathbf{x} \triangleq \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad {}^b\mathbf{R}_1 = e^{-\delta_0\,[\mathbf{z}^\wedge]} \tag{3.21} $$

where $\mathbf{S}_\mathbf{x}$ accounts for the fact that the end effector z-coordinate appears in the second element of $\mathbf{x}_p$.

Also, $\delta\mathbf{k}_\kappa$ affects the orientation of ${}^b\mathbf{R}_g$ by inducing $\delta\theta_e$:

$$ {}^b\delta\boldsymbol{\mu}_\kappa = -\hat{\mathbf{y}}_1\mathbf{S}_\mu\delta\mathbf{x}_p, \quad \mathbf{S}_\mu \triangleq \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \tag{3.22} $$

where $\mathbf{S}_\mu$ accounts for θ_e appearing in the last element of $\mathbf{x}_p$.

Substituting the error propagation of $\delta\mathbf{x}_p = \mathbf{J}_{\mathbf{k}_\kappa}\delta\mathbf{k}_\kappa$ to Eq. (3.21) and Eq. (3.22), we obtain the error propagation due to $\delta\mathbf{k}_\kappa$:

$$ {}^b\delta\boldsymbol{\xi}(\delta\mathbf{k}_\kappa) = \begin{bmatrix} {}^b\mathbf{R}_1\,\mathbf{S}_\mathbf{x}\,\mathbf{J}_{\mathbf{k}_\kappa} \\ \hdashline -\hat{\mathbf{y}}_1\,\mathbf{S}_\mu\,\mathbf{J}_{\mathbf{k}_\kappa} \end{bmatrix} \delta\mathbf{k}_\kappa \tag{3.23} $$

3.3.3 Pose Error due to Twist Deviation $\left(\delta\xi(\delta\mathbf{k}_\gamma)\right)$

The twist parameter $\mathbf{k}_\gamma$ does not affect the end-effector position. Therefore, only the gripper frame orientation ${}^b\mathbf{R}_{g_s}(L)$ is considered. Using the definition of $\mathbf{k}_\gamma$ and $\mathbf{k}_\kappa$ in Eq. (3.12) we obtain the functional relationship:

$$ {}^b\mathbf{R}_{g_s}(s) = \mathbf{R}(\psi, s, \mathbf{k}_\kappa, \mathbf{k}_\gamma) \tag{3.24} $$

The estimation of $\mathbf{k}_\kappa$ is described in 3.5.1 using the identification Jacobian $\mathbf{J}_{\mathbf{k}_\kappa}$. With $\mathbf{k}_\kappa$ estimated and assumed fixed, we derive the identification Jacobian $\mathbf{J}_{\mathbf{k}_\gamma}$ relating $\delta\mathbf{k}_\gamma$ to its corresponding differential rotation of $\{\mathrm{G}\}$.

To simplify notation, we will use the shorthand notation $\mathbf{R} \triangleq {}^{b}\mathbf{R}_{gs}(s)$. Following [63], the differentiation of the rotation matrix, $\mathbf{R}$, given $\delta\mathbf{k}_{\gamma}$, is expressed as:

$$\mathrm{d}\mathbf{R} = \frac{\partial \mathbf{R}}{\partial \gamma_{\mathrm{e}0}}\delta\gamma_{\mathrm{e}0} + \frac{\partial \mathbf{R}}{\partial \mathrm{k}_{\theta}}\delta\mathrm{k}_{\theta} + \frac{\partial \mathbf{R}}{\partial \mathrm{k}_{\delta}}\delta\mathrm{k}_{\delta} \tag{3.25}$$

Applying the differentiation chain rule, we arrive at:

$$\mathrm{d}\mathbf{R} = \frac{\partial \mathbf{R}}{\partial \gamma(\mathrm{s})}\left(\frac{\partial \gamma(\mathrm{s})}{\partial \gamma_{\mathrm{e}0}}\delta\gamma_{\mathrm{e}0} + \frac{\partial \gamma(\mathrm{s})}{\partial \mathrm{k}_{\theta}}\delta\mathrm{k}_{\theta} + \frac{\partial \gamma(\mathrm{s})}{\partial \mathrm{k}_{\delta}}\delta\mathrm{k}_{\delta}\right) \tag{3.26}$$

where $\frac{\partial \mathbf{R}}{\partial \gamma(s)}$ can be derived as (details see appendix B.3):

$$\frac{\partial \mathbf{R}}{\partial \gamma(s)} = \mathbf{R}\left[\mathbf{e}_3{}^{\wedge}\right], \quad \left[\mathbf{e}_3{}^{\wedge}\right] = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{3.27}$$

We next express $\mathrm{d}\mathbf{R}$ as a sequence of small rotations $\delta\alpha_x$, $\delta\alpha_y$ and $\delta\alpha_z$ about axes of the gripper. Following [72] we obtain:

$$\mathrm{d}\mathbf{R} = \mathbf{R}\delta\mathbf{R}, \quad \delta\mathbf{R} = \begin{bmatrix} 0 & -\delta\alpha_z & \delta\alpha_y \\ \delta\alpha_z & 0 & -\delta\alpha_x \\ -\delta\alpha_y & \delta\alpha_x & 0 \end{bmatrix} \tag{3.28}$$

Substituting $\mathrm{d}\mathbf{R}$ from Eq. (3.26) in Eq. (3.28) and solving for $\delta\mathbf{R}$ gives:

$$\delta\mathbf{R} = \begin{bmatrix} 0 & -\left(\frac{\partial \gamma(s)}{\partial \mathbf{k}_{\gamma}}\right)^{\mathrm{T}}\delta\mathbf{k}_{\gamma} & 0 \\ \left(\frac{\partial \gamma(s)}{\partial \mathbf{k}_{\gamma}}\right)^{\mathrm{T}}\delta\mathbf{k}_{\gamma} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{3.29}$$

The skew symmetric matrix $\delta\mathbf{R}$ is expressed in the moving gripper frame and its corresponding

vector form is ${}^{g}\boldsymbol{\delta\mu}_{\gamma}$:

$$ {}^{g}\boldsymbol{\delta\mu}_{\gamma} \triangleq \begin{bmatrix} \mathbf{0}_{2\times 1} \\ \left(\frac{\partial \boldsymbol{\gamma}(s)}{\partial \mathbf{k}_{\gamma}}\right)^{\mathrm{T}} \delta\mathbf{k}_{\gamma} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{2\times 3} \\ \left(\frac{\partial \boldsymbol{\gamma}(s)}{\partial \mathbf{k}_{\gamma}}\right)^{\mathrm{T}} \end{bmatrix} \boldsymbol{\delta\mathbf{k}}_{\gamma} \tag{3.30} $$

Using ${}^{b}\boldsymbol{\delta\mu}_{\gamma} = {}^{b}\mathbf{R}_{g}\,{}^{g}\boldsymbol{\delta\mu}_{\gamma}$ to convert ${}^{g}\boldsymbol{\delta\mu}_{\gamma}$ to the base frame $\{\mathrm{B}\}$, we obtain the *twist identification* Jacobian as:

$$ {}^{b}\boldsymbol{\delta\mu}_{\gamma} \triangleq \mathbf{J}_{\mathbf{k}_{\gamma}}(s)\,\boldsymbol{\delta\mathbf{k}}_{\gamma}, \quad \mathbf{J}_{\mathbf{k}_{\gamma}}(s) = {}^{b}\mathbf{R}_{g} \begin{bmatrix} \mathbf{0}_{2\times 3} \\ \left(\frac{\partial \boldsymbol{\gamma}(s)}{\partial \mathbf{k}_{\gamma}}\right)^{\mathrm{T}} \end{bmatrix} \tag{3.31} $$

Finally, using the definition of differential pose vector as in Eq. (3.20), we express the effect of $\boldsymbol{\delta\mathbf{k}}_{\gamma}$ on the end effector pose as:

$$ {}^{b}\boldsymbol{\delta\xi}(\boldsymbol{\delta\mathbf{k}}_{\gamma}) = \begin{bmatrix} \mathbf{0}_{3\times 3} \\ \mathbf{J}_{\mathbf{k}_{\gamma}} \end{bmatrix} \boldsymbol{\delta\mathbf{k}}_{\gamma} \tag{3.32} $$

3.3.4 Kinematic Error Propagation for a Multi-segment Robot

The pose error model for a single segment is achieved by substituting Eq. (3.23) and Eq. (3.32) in Eq. (3.17) and vectorizing:

$$ {}^{b}\boldsymbol{\delta\xi}(\boldsymbol{\delta\mathbf{k}}_{T}) = \left[\begin{array}{c:c} {}^{b}\mathbf{R}_{1}\,\mathbf{S}_{\mathbf{x}}\,\mathbf{J}_{\mathbf{k}_{\kappa}} & \mathbf{0}_{3\times 3} \\ \hdashline -\hat{\mathbf{y}}_{1}\,\mathbf{S}_{\mu}\,\mathbf{J}_{\mathbf{k}_{\kappa}} & \mathbf{J}_{\mathbf{k}_{\gamma}} \end{array}\right] \begin{bmatrix} \boldsymbol{\delta\mathbf{k}}_{\kappa} \\ \boldsymbol{\delta\mathbf{k}}_{\gamma} \end{bmatrix} \tag{3.33} $$

Defining the *task space segment* identification Jacobian $\mathbf{J}_{\mathbf{k}_{T}}$ such that ${}^{b}\boldsymbol{\delta\xi}(\boldsymbol{\delta\mathbf{k}}_{T}) = \mathbf{J}_{\mathbf{k}_{T}}\boldsymbol{\delta\mathbf{k}}_{T}$, we obtain:

$$ \mathbf{J}_{\mathbf{k}_{T}} = \begin{bmatrix} {}^{b}\mathbf{R}_{1}\,\mathbf{S}_{\mathbf{x}}\,\mathbf{J}_{\mathbf{k}_{\kappa}} & \mathbf{0}_{3\times 3} \\ -\hat{\mathbf{y}}_{1}\,\mathbf{S}_{\mu}\,\mathbf{J}_{\mathbf{k}_{\kappa}} & \mathbf{J}_{\mathbf{k}_{\gamma}} \end{bmatrix} \triangleq \left[\begin{array}{c} \mathbf{J}_{\mathbf{k}_{\mathbf{x}}} \\ \hdashline \mathbf{J}_{\mathbf{k}_{\mu}} \end{array}\right] \tag{3.34} $$

We next consider the error propagation of an n-segment continuum robot. We use

$\mathbf{J}_{\mathbf{k}_{T_i}}$ and $\mathbf{k}_{T_i}$ to denote the identification Jacobian and the calibration parameters vector and $\{B_i\}$ and $\{E_i\}$ as the base and end-effector frames of the i^{th} segment. The error propagation problem of a multi-segment continuum robot is equivalent to its geometric Jacobian formulation [89]: one can think of the robot pose error $\delta\boldsymbol{\xi}$ as the tip "twist" vector due to "joint" speeds $\delta\mathbf{k}_T$.

The *relative* pose error of the i^{th} segment, ${}^{b_i}\delta\boldsymbol{\xi}_{e_i/b_i}$, is only affected by its own parameters $\delta\mathbf{k}_{T_i}$ and it is obtained as:

$$ {}^{b_i}\delta\boldsymbol{\xi}_{e_i/b_i} = \mathbf{J}_{\mathbf{k}_{T_i}}\delta\mathbf{k}_{T_i} \tag{3.35} $$

To consider the *absolute* pose error of the i^{th} segment, ${}^{0}\delta\boldsymbol{\xi}_{e_i/0}$, we have to account for the translations and rotations of all segments proximal to the i^{th} segment[5]. Using frame transformation of twist, we obtain:

$$ {}^{0}\delta\boldsymbol{\xi}_{e_i/0} = \sum_{j=1}^{i}\left(\mathbf{S}_{ji}\,{}^{b_j}\delta\boldsymbol{\xi}_{e_j/b_j}\right) \tag{3.36} $$

$$ \mathbf{S}_{ji} = \begin{bmatrix} {}^{0}\mathbf{R}_{b_j} & \left[{}^{0}\mathbf{x}_{e_j/e_i}\right]^{\wedge}\,{}^{0}\mathbf{R}_{b_j} \\ \mathbf{0} & {}^{0}\mathbf{R}_{b_j} \end{bmatrix}, \quad j \in [1,i] \tag{3.37} $$

where $\left[{}^{0}\mathbf{x}_{e_j/e_i}\right]^{\wedge}$ denotes the cross product matrix of the vector ${}^{0}\mathbf{x}_{e_j/e_i}$ pointing from e_i to e_j and expressed in $\{0\}$:

$$ {}^{0}\mathbf{x}_{e_j/e_i} \triangleq {}^{0}\mathbf{x}_{e_j} - {}^{0}\mathbf{x}_{e_i}, \quad j \in [1,i] \tag{3.38} $$

We note that if the calibration sensor of the i^{th} segment is not at its end disk, but as a known arc length, then the equations above still hold as long as one interprets e_i to be the location of the sensor on the i^{th} segment.

Rewriting Eq. (3.36) in vector format to obtain the error propagation of the i^{th} segment

[5]Frame$\{0\}$ is the base of the 1$^{\text{st}}$ segment, i.e. $\{B_1\}$

results in:

$$ {}^{0}\delta\boldsymbol{\xi}_{e_i/0} = \left[\begin{array}{c:c:c} \mathbf{S}_{1i}\mathbf{J}_{\mathbf{k}_{T_1}} & \cdots & \mathbf{S}_{ii}\mathbf{J}_{\mathbf{k}_{T_i}} \end{array} \right] \begin{bmatrix} \delta\mathbf{k}_{T_1} \\ \vdots \\ \delta\mathbf{k}_{T_i} \end{bmatrix} \tag{3.39} $$

For the purpose of updating all the parameters of an n-segment continuum robot simultaneously, we define an *augmented*[6] parameter vector $\widehat{\mathbf{k}}_T$ as:

$$ \widehat{\mathbf{k}}_T \triangleq \left[(\mathbf{k}_{T_i})^{\mathrm{T}}, \ldots, (\mathbf{k}_{T_n})^{\mathrm{T}} \right]^{\mathrm{T}}, \quad \widehat{\mathbf{k}}_T \in \mathbb{R}^{8n\times 1} \tag{3.40} $$

We also define an augmented differential pose vector as $\delta\widehat{\boldsymbol{\xi}}$:

$$ \delta\widehat{\boldsymbol{\xi}} \triangleq \left[{}^{0}\delta\boldsymbol{\xi}_{e_1/0}^{\mathrm{T}}, \ldots, {}^{0}\delta\boldsymbol{\xi}_{e_n/0}^{\mathrm{T}} \right]^{\mathrm{T}}, \quad \delta\widehat{\boldsymbol{\xi}} \in \mathbb{R}^{6n\times 1} \tag{3.41} $$

The augmented identification Jacobian is then obtained as:

$$ \delta\widehat{\boldsymbol{\xi}} \triangleq \widehat{\mathbf{J}}_{\mathbf{k}_T}\delta\widehat{\mathbf{k}}_T, \quad \widehat{\mathbf{J}}_{\mathbf{k}_T} \in \mathbb{R}^{6n\times 8n} \tag{3.42} $$

$$ \widehat{\mathbf{J}}_{\mathbf{k}_T} = \begin{bmatrix} \mathbf{S}_{11}\mathbf{J}_{\mathbf{k}_{T_1}} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{S}_{12}\mathbf{J}_{\mathbf{k}_{T_1}} & \mathbf{S}_{22}\mathbf{J}_{\mathbf{k}_{T_2}} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \mathbf{0} \\ \mathbf{S}_{1n}\mathbf{J}_{\mathbf{k}_{T_1}} & \mathbf{S}_{2n}\mathbf{J}_{\mathbf{k}_{T_2}} & \cdots & \mathbf{S}_{nn}\mathbf{J}_{\mathbf{k}_{T_n}} \end{bmatrix} \tag{3.43} $$

The augmented identification Jacobian $\widehat{\mathbf{J}}_{\mathbf{k}_T}$ is used in section 3.5.4 to estimate the parameter vector $\widehat{\mathbf{k}}_T$.

3.4 Kinematic Error Propagation from Configuration to Joint Space

The inverse and direct kinematics problems for a continuum segment without twist have closed-form solutions as presented in [88]. However, in the presence of twist and

[6] We will henceforth use $\widehat{(\cdot)}$ to denote *"augmented"* vectors representing concatenated entities for a multi-segment robot.

shape deviation such solutions are attainable only via numerical methods. In this section, we derive the inverse and direct kinematics and an associated identification Jacobian.

3.4.1 Inverse Kinematics

In addition to previously defined nomenclature, we define $\mathbf{v}(s)$ and $\mathbf{w}(s)$ to facilitate the derivation of the kinematics mapping with backbone twist and bending shape deviation. Figure 3.4 shows these two vectors. The vector $\mathbf{v}(s)$ denotes the radial vector pointing from the central backbone at arc-length coordinate s to the first secondary backbone (it is aligned with $\hat{\mathbf{x}}_{g_s}$). Vector $\mathbf{w}(s)$ is the position vector of the point on the first secondary backbone relative to the origin of $\{\mathrm{B}\}$. The inverse kinematics solves the required secondary backbone lengths $L_i, i = 1,2,3$ for a given configuration space vector $\psi = (t, \delta_0)$. The length of the i^{th} secondary backbone, L_i, can be calculated using the tangential vector $\mathrm{d}\mathbf{w}(s)/\mathrm{d}s$.

$$L_i = \int_0^L \left\| \frac{\mathrm{d}\mathbf{w}(s)}{\mathrm{d}s} \right\| \mathrm{d}s, \quad \frac{\mathrm{d}\mathbf{w}(s)}{\mathrm{d}s} = \frac{\mathrm{d}\mathbf{x}(s)}{\mathrm{d}s} + \frac{\mathrm{d}\mathbf{v}(s)}{\mathrm{d}s} \tag{3.44}$$

where $\mathbf{x}(\mathbf{s})$ is according to Eq. (3.10) and then $\frac{\mathrm{d}\mathbf{x}(s)}{\mathrm{d}s}$ is easily obtained. Using Eq. (3.11) and Eq. (3.6), and referring to Fig. 3.4, we derive the angular velocity of frame $\{\mathrm{Gs}\}$ (denoted $\boldsymbol{\omega}_{gs}$) and the velocity of the tip of $\mathbf{v}$ as follows:

$$^1\left(\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}s}\right) = {}^1\mathbf{R}_{e_s}\,({}^{e_s}\boldsymbol{\omega}_{g_s} \times {}^{e_s}\mathbf{v}) \tag{3.45}$$

$${}^{e_s}\boldsymbol{\omega}_{g_s} = \hat{\mathbf{z}}\frac{\mathrm{d}\gamma(s)}{\mathrm{d}s} - \hat{\mathbf{y}}\frac{\mathrm{d}\theta(s)}{\mathrm{d}s} = \left[0, -\kappa(s,t), \frac{\gamma_e}{L}\right]^{\mathrm{T}} \tag{3.46}$$

$${}^1\mathbf{R}_{e_s} = e^{(\frac{\pi}{2}-\theta(s))\,[\mathbf{e}_2{}^\wedge]}, \quad {}^{e_s}\mathbf{v} = r\left[c_{\sigma_i(s)}, s_{\sigma_i(s)}, 0\right]^{\mathrm{T}} \tag{3.47}$$

where $\sigma_i(s)$ designates the angular coordinate of the i'th backbone given by:

$$\sigma_i(s) = \delta(s) + (i-1)\beta \tag{3.48}$$

and r denotes the constant radial distance between the central and secondary backbones.

By substituting the expressions of $\frac{\mathrm{d}\mathbf{x}(s)}{\mathrm{d}s}$ and $\frac{\mathrm{d}\mathbf{v}(s)}{\mathrm{d}s}$ into Eq. (3.44) we obtain the length of the i^{th} backbone:

$$L_i = \int_0^L g_i(s)\,\mathrm{d}s, \qquad i \in [1,3] \tag{3.50}$$

$$g_i(s) = \sqrt{\left(r\kappa(s,t)c_{\sigma_i(s)} + 1\right)^2 + \left(\tfrac{r\gamma_e}{L}\right)^2} \tag{3.51}$$

The *joint* space vector, $\mathbf{q} = [q_1, q_2, q_3]^{\mathrm{T}}$, is the push/pull displacement of the backbones, i.e. $q_i = L_i - L$. Defining the "home" position $\mathbf{q}_h = \left[q_{h_1}, q_{h_2}, q_{h_3}\right]^{\mathrm{T}}$, for a straight segment completes the formulation of the inverse kinematics problem:

$$\mathbf{q} = \mathbf{f}_q(\boldsymbol{\psi}, \mathbf{k}), \quad f_{q_i} = L_i(\boldsymbol{\psi}, \mathbf{k}) - L + q_{h_i}, \quad i \in [1,3] \tag{3.52}$$

where $\mathbf{k}$ is a kinematic model parameter vector defined as:

$$\mathbf{k} \triangleq \left[\mathbf{k}_\kappa^{\mathrm{T}}, \mathbf{k}_\gamma^{\mathrm{T}}, \mathbf{k}_\mathbf{q}^{\mathrm{T}}\right]^{\mathrm{T}}, \quad \mathbf{k}_\mathbf{q} \triangleq [\mathbf{q}_h^{\mathrm{T}}, r]^{\mathrm{T}} \tag{3.53}$$

The *overall* calibration parameter vector $\mathbf{k}$ includes a *Joint Characteristic* calibration vector $\mathbf{k}_\mathbf{q}$, which captures modeling errors in joint home positions and backbone radial distance.

3.4.2 Direct Kinematics

Algorithm 2 presents an iterative solver for configuration space direct kinematics based on Newton's method. This method requires the *configuration* space kinematic Jacobian, $\mathbf{J}_{\mathbf{q}\psi}$, which relates infinitesimal $\delta\boldsymbol{\psi}$ to $\delta\mathbf{q}$, i.e. $\mathbf{J}_{\mathbf{q}\psi}\delta\boldsymbol{\psi} = \delta\mathbf{q}$. This $\mathbf{J}_{\mathbf{q}\psi}$ can be derived from Eqs. (3.52, 3.50, 3.51):

$$\mathbf{J}_{\mathbf{q}\psi} = \frac{\partial \mathbf{f}_q(\boldsymbol{\psi}, \mathbf{k})}{\partial \boldsymbol{\psi}} = \int_0^L \frac{\partial \mathbf{g}(s, \boldsymbol{\psi}, \mathbf{k})}{\partial \boldsymbol{\psi}}\,\mathrm{d}s \tag{3.54}$$

where $\mathbf{J}_{\mathbf{q}\psi}$ and $\frac{\partial g_i}{\partial \psi_j}$ is obtained by differentiation of Eq. (3.51).

Algorithm 2 does not guarantee convergence to a global minimum, but the two-dimensional search space for a given continuum segment facilitates a multi-start computation to minimize the chance of missing the global minimum. This algorithm is used only for kinematic simulations for the purposes of visualization and evaluation of calibration results.

Algorithm 2 Configuration Space Direct Kinematics

Require: $\{\mathbf{q}_{tg},\ \mathbf{k}\}$

Require: ψ_0, $\varepsilon_0 > 0$, $\mu > 0$

1: **START** Initialize: $\psi \leftarrow \psi_0$, $e_q \leftarrow 100\,\varepsilon_0$

2: **while** $e_q > \varepsilon_0$ **do**

3: Compute $\mathbf{q} = \mathbf{f}(\psi, \mathbf{k})$, $\mathbf{J}_{\mathbf{q}\psi} = \mathbf{J}_{\mathbf{q}\psi}(\psi, \mathbf{k})$ using Eq. (3.52) and Eq. (3.54).

4: Update $\{\psi\}$

$$\mathbf{J}^{+}_{\mathbf{q}\psi} = (\mathbf{J}^{T}_{\mathbf{q}\psi}\mathbf{W}\mathbf{J}_{\mathbf{q}\psi})^{-1}\mathbf{J}^{T}_{\mathbf{q}\psi}\mathbf{W},\ \ \Delta\psi = +\mathbf{J}^{+}_{\mathbf{q}\psi}\Delta\mathbf{q}, \tag{3.55}$$

5: $\Delta\mathbf{q} \leftarrow (\mathbf{q}_{tg} - \mathbf{q})$, $\psi \leftarrow (\psi + \mu\Delta\psi)$, $e_q \leftarrow \|\Delta\mathbf{q}\|$

6: **end while**

7: $\psi_{tg} \leftarrow \psi$ **END**

Ensure: ψ_{tg}

3.4.3 Joint Space Identification Jacobian

The multi-segment identification Jacobian in Eq. (3.42) relates shape and twist parameter errors to task space errors and allows identification of $\mathbf{k}_\kappa$ and $\mathbf{k}_\gamma$ but it does not capture the effect of homing and radial offsets or allow the identification of $\mathbf{k}_q$. We therefore present an identification Jacobian $\mathbf{J}_\mathbf{k}$ allowing for identifying the overall calibration parameter vector $\mathbf{k}$ as defined in Eq. (3.53). In subsequent sections we evaluate the utility and limitations of the use of this identification Jacobian.

We next define $\mathbf{J}_\mathbf{k}$ relating errors in $\mathbf{k}$ to errors in joint variables $\mathbf{q}$, i.e. $\delta\mathbf{q} = \delta\mathbf{f}_\mathbf{q}(\psi, \mathbf{k}) =$

$\mathbf{J_k}\delta\mathbf{k}$. Referring to Eq. (3.52) and Eq. (3.50) and using the Leibniz rule results in:

$$\begin{aligned}\mathbf{J_k} = \frac{\partial \mathbf{f_q}(\psi,\mathbf{k})}{\partial \mathbf{k}} = \int_0^L \frac{\partial \mathbf{g}(s,\psi,\mathbf{k})}{\partial \mathbf{k}}\,\mathrm{d}s \\ + \mathbf{g}(L,\psi,\mathbf{k})\frac{\partial L}{\partial \mathbf{k}} - \frac{\partial [L,L,L]^T}{\partial \mathbf{k}}\end{aligned} \tag{3.56}$$

where $\mathbf{J_k} \in \mathbb{R}^{3\times(8+2(m+1))}$, $\mathbf{g}(s,\psi,\mathbf{k}) = [g_1,g_2,g_3]^T$, $\frac{\partial g_i}{\partial k_j}$ is obtained by differentiating Eq. (3.51) and m is the highest degree in $\boldsymbol{\eta}(s)$ defined in Eq. (3.5).

3.5 Estimation of Calibration Parameters

Sections (3.3, 3.4) presented the kinematic models from *Configuration* space to *Task* space and from *Configuration* space to *Joint* along with associated identification Jacobians relating errors in kinematic variables ($\{\delta\psi, \delta\mathbf{x}, \delta\mathbf{q}\}$) to errors in kinematic model parameters ($\{\delta\mathbf{k}_\kappa, \delta\mathbf{k}_\gamma, \delta\mathbf{k_q}\}$). Through this section we discuss the method and formulation to estimate/calibrate these parameters.

Recall from sections ((3.3, 3.4) that the parameter vectors $\{\mathbf{k}_\kappa, \mathbf{k}_\gamma, \mathbf{k_q}\}$ are dependent in both of the *configuration* to *task* space kinematic model and the *configuration* to *joint* space kinematic model, meaning that we can use either model formulation and the corresponding measurements to estimate the parameter values. This provides different choices for methods, algorithms and observations. Hence in this section, different types of observations/measurements, different corresponding error functions, and different objective cost functions are discussed for different cases. Despite these differences, all calibration/estimation algorithms used in this section rely on nonlinear least-square estimation.

3.5.1 Capturing the Shape Characteristic Vector - $\mathbf{k}_\kappa$

In this case, six-axis position and orientation sensor(s) is(are) assumed to be installed on the primary backbone. The position measurements[7] will be directly used while the

[7] We will henceforth use $\bar{(\cdot)}$ to designate measured data.

orientation measurements will be converted to a measurement $\bar{t}$ using (3.2) as the following:

$$\bar{t} = \left(\bar{\theta}_e - \bar{\theta}_{e_a}\right) / \left(\bar{\theta}_{e_b} - \bar{\theta}_{e_a}\right) \tag{3.57}$$

where $\{\bar{\theta}_{e_b}, \bar{\theta}_{e_a}\}$ denote the end-effector angles for the two shape generators as depicted in Fig. 3.2(a) and defined in Eq. (3.3).

We next assume that N configurations are used for calibration. For each configuration, we measure $\bar{\mathbf{x}}_{p_j}, j \in [1,N]$ as defined in Eq. (3.13). At the j^{th} pose and for an assumed shape characteristic vector $\mathbf{k}_\kappa$, we define the error vector $\mathbf{c}_{p_j}$:

$$\mathbf{c}_{p_j} \triangleq \bar{\mathbf{x}}_{p_j} - \mathbf{f}_{p_j}(\bar{t}_j, \mathbf{k}_\kappa) = \mathbf{c}_{p_j}(\mathbf{k}_\kappa), \quad {}_{j=1,\ldots,N} \tag{3.58}$$

We also define the *aggregated*[8] error over all poses as:

$$\widetilde{\mathbf{c}}_p(\mathbf{k}_\kappa) \triangleq \widetilde{\mathbf{x}}_p - \widetilde{\mathbf{f}}_p(\mathbf{k}_\kappa), \quad \widetilde{\mathbf{c}}_p,\, \widetilde{\mathbf{x}}_p,\, \widetilde{\mathbf{f}}_p \in \mathbb{R}^{3N\times 1} \tag{3.59}$$

$$\widetilde{\mathbf{x}}_p \triangleq \left[\bar{\mathbf{x}}_{p_1}{}^{\mathrm{T}}, \ldots, \bar{\mathbf{x}}_{p_N}{}^{\mathrm{T}}\right]^{\mathrm{T}}, \quad \widetilde{\mathbf{f}}_p \triangleq \left[\mathbf{f}_{p_1}{}^{\mathrm{T}}, \ldots, \mathbf{f}_{p_N}{}^{\mathrm{T}}\right]^{\mathrm{T}} \tag{3.60}$$

and define an objective function M_p quantifying the aggregated error over the calibration workspace:

$$M_p(\mathbf{k}_\kappa) = \tfrac{1}{2N}\, \widetilde{\mathbf{c}}_p{}^{\mathrm{T}}\, \mathbf{W}\, \widetilde{\mathbf{c}}_p \tag{3.61}$$

where $\mathbf{W}$ is a diagonal positive definite weight matrix encoding relative confidence in the measurement accuracy.

The first-order Taylor series approximation of M_p is:

$$M_p(\mathbf{k}_\kappa + \delta\mathbf{k}_\kappa) \approx M_p(\mathbf{k}_\kappa) + \mathbf{J}_{M_p}\delta\mathbf{k}_\kappa \tag{3.62}$$

[8]We will henceforth use $\widetilde{(\cdot)}$ to designate *"aggregated"* entities representing concatenated data from all calibration poses.

where *the aggregated* Jacobian $\mathbf{J}_{M_p} \in \mathbb{R}^{1\times 5}$ is given by:

$$\mathbf{J}_{M_p} = \tfrac{1}{N}(\widetilde{\mathbf{c}}_p)^{\mathrm{T}}\,\mathbf{W}\,\mathbf{J}_{c_p} \tag{3.63}$$

$$\mathbf{J}_{c_p} = \frac{\partial \widetilde{\mathbf{c}}_p}{\partial \mathbf{k}_\kappa} = -\left[\left(\frac{\partial \mathbf{f}_{p_1}}{\partial \mathbf{k}_\kappa}\right)^{\mathrm{T}}, \ldots, \left(\frac{\partial \mathbf{f}_{p_N}}{\partial \mathbf{k}_\kappa}\right)^{\mathrm{T}}\right]^{\mathrm{T}} \tag{3.64}$$

Equation (3.63) shows that minimizing M_p entails following the gradient descent direction along $\frac{\partial \widetilde{\mathbf{c}}_p}{\partial \mathbf{k}_\kappa}$. In algorithm 3, $\mathbf{H}$ is the parameter scaling matrix discussed in section 3.5.5. The task space variable scaling is achieved by adjusting $\mathbf{W}$, [85].

Algorithm 3 Nonlinear LS Estimate

Require: $\mathscr{D}\{(\mathbf{p}_j, \psi_j)\}_{,\, j=1,\ldots,N}$; $\mathbf{k}_{\kappa_0}$, $(\beta, \eta) > 0$

1: **START** Initialize: $\mathbf{k}_i \leftarrow \mathbf{k}_{\kappa_0}$, $M_{i-1} \leftarrow 1$, $M_i \leftarrow 100$

2: **while** $\frac{\|M_{p_i} - M_{p_{i-1}}\|}{M_{p_{i-1}}} \geq \beta$ **do**

3: $\quad M_{p_{i-1}} \leftarrow M_{p_i}$, $\widetilde{\mathbf{c}}_p = \widetilde{\mathbf{c}}_p(\mathbf{k}_i)$, $M_{p_i} = M_{p_i}(\mathbf{k}_i)$,

4: $\quad \mathbf{J}_{c_p} = \mathbf{J}_{c_p}(\mathbf{k}_i)$

5: $\quad$ Update $\mathbf{k}_{i+1}$:

$$\mathbf{k}_{i+1} = \mathbf{k}_i - \mathbf{H}\left(\eta\,\left(\mathbf{J}_{c_p}\right)^{+}\widetilde{\mathbf{c}}_p\right), \eta \in (0,1] \tag{3.65}$$

$$\left(\mathbf{J}_{c_p}\right)^{+} = \left((\mathbf{J}_{c_p})^{\mathrm{T}}\mathbf{W}\mathbf{J}_{c_p}\right)^{-1}(\mathbf{J}_{c_p})^{\mathrm{T}}\mathbf{W} \tag{3.66}$$

6: **end while**

7: $\mathbf{k}^* \leftarrow \mathbf{k}_i$

Ensure: $\mathbf{k}^*$

3.5.2 Capturing the Twist Characteristic Vector - $\mathbf{k}_\gamma$

In this case, the sensor is assumed to be placed either at the end-disk or at arc-length $l \in (0, L]$ along the primary backbone. A measured sensor orientation $\bar{\mathbf{R}}_j$ is assumed available, where subscript j denotes the j^{th} measurement. A measurement of the configuration space is also assumed available $\bar{\psi}_j = \left[\bar{t}_j, \bar{\delta}_j\right]^{\mathrm{T}}$, where $\bar{t}_j$ can be recalled from Eq. (3.57) and $\bar{\delta}$ can be calculated from the sensor measurement. Using the kinematic model, and given an estimate of the model parameter values $\{\bar{\mathbf{k}}_\kappa, \bar{\mathbf{k}}_\gamma\}$, the modeled sensor orientation, $\mathbf{R}_j \triangleq {}^{b}\mathbf{R}_{g_s}(l)$, can be obtained from Eq. (3.24).

We next capture the orientation error between the measured orientation $\bar{\mathbf{R}}_j$ and the

model-predicted orientation $\mathbf{R}_j$. Using a fixed-frame rotation sequence, this error is calculated as:

$$\mathbf{R}_{e_j} \triangleq \bar{\mathbf{R}}_j \mathbf{R}_j{}^{\mathrm{T}} = e^{\alpha_{e_j}[\hat{\mathbf{m}}_{e_j}]^\wedge} \tag{3.67}$$

where α_{e_j} and $\hat{\mathbf{m}}_{e_j}$ are the angle and axis parameterizing the orientation error $\mathbf{R}_{e_j}$. These parameters are given by:

$$\alpha_{e_j} = \cos^{-1}\left(\frac{\mathrm{Tr}(\mathbf{R}_{e_j})-1}{2}\right) \tag{3.68}$$

$$\hat{\mathbf{m}}_{e_j} = \frac{1}{2\sin(\alpha_{e_j})}\left(\mathbf{R}_{e_j} - \mathbf{R}_{e_j}{}^{\mathrm{T}}\right)^\vee \tag{3.69}$$

where the operator $(\cdot)^\vee$ designates the vector of a skew-symmetric matrix.

For calibration formulation, we parameterize the orientation error using an orientation error vector $\mathbf{c}_{\gamma_j}$ defined as:

$$\mathbf{c}_{\gamma_j} = \alpha_{e_j}\hat{\mathbf{m}}_{e_j}, \qquad \mathbf{c}_{\gamma_j} \in \mathbb{R}^3 \tag{3.70}$$

As in the previous section, an aggregated error vector $\widetilde{\mathbf{c}}_\gamma$ is defined as $\widetilde{\mathbf{c}}_\gamma = \left[(\mathbf{c}_{\gamma_1})^{\mathrm{T}}, \ldots, (\mathbf{c}_{\gamma_N})^{\mathrm{T}}\right]^{\mathrm{T}}$ and the objective function M_γ is defined as:

$$M_\gamma(\mathbf{k}_\kappa, \mathbf{k}_\gamma) = \tfrac{1}{2N}\widetilde{\mathbf{c}}_\gamma{}^{\mathrm{T}}\,\mathbf{W}\,\widetilde{\mathbf{c}}_\gamma \tag{3.71}$$

Following a similar derivation as in the previous section, we define an *aggregated Jacobian* $\mathbf{J}_{M_\gamma} = \frac{1}{N}(\widetilde{\mathbf{c}}_\gamma)^{\mathrm{T}}\,\mathbf{W}\,\mathbf{J}_{c_\gamma}$ where $\mathbf{J}_{c_\gamma}$ is obtained using the relation $\delta\mathbf{c}_{\gamma_j} = -\mathbf{J}_{\mathbf{k}_\gamma}\delta\mathbf{k}_\gamma$ as shown in Appendix C.1 and $\mathbf{J}_{k_\gamma}$ was defined in Eq. (3.31).

$$\mathbf{J}_{c_\gamma} \triangleq \frac{\partial\widetilde{\mathbf{c}}_\gamma}{\partial\mathbf{k}_\gamma} = -\left[\left(\mathbf{J}_{\mathbf{k}_\gamma(\bar{\psi}_1)}\right)^{\mathrm{T}}, \ \cdots, \ \left(\mathbf{J}_{\mathbf{k}_\gamma(\bar{\psi}_N)}\right)^{\mathrm{T}}\right]^{\mathrm{T}} \tag{3.72}$$

Finally, $\mathbf{k}_\gamma$ is estimated using nonlinear least squares.

3.5.3 *Capturing the Overall Characteristic Vector* - **k**

In section 3.7.3 we evaluate the premise of directly estimating the overall kinematic parameter vector **k** previously defined in Eq. (3.53). We therefore derive a method for estimating **k** from N measurements $\left(\bar{\mathbf{q}}_j, \bar{\psi}_j\right), j \in [1 \ldots N]$. We first define a joint-space error vector based on the numerical inverse kinematics solution from section 3.4.1:

$$\mathbf{c}_{q_j}(\mathbf{k}) \triangleq \bar{\mathbf{q}}_j - \mathbf{f}_{q_j}(\bar{\psi}_j, \mathbf{k}) \tag{3.73}$$

where $\mathbf{f}_q$ was defined in Eq. (3.52) irrespective of the pose j.

To capture the joint-space error over the calibration data set we define the aggregated measured joint vector $\widetilde{\bar{\mathbf{q}}}$ and the aggregated model-based joint vector $\widetilde{\mathbf{f}}_q$ as:

$$\widetilde{\bar{\mathbf{q}}} \triangleq \left[\bar{\mathbf{q}}_1{}^{\mathrm{T}}, \ldots, \bar{\mathbf{q}}_N{}^{\mathrm{T}}\right]^{\mathrm{T}}, \quad \widetilde{\mathbf{f}}_q \triangleq \left[\mathbf{f}_{q_1}{}^{\mathrm{T}}, \ldots, \mathbf{f}_{q_N}{}^{\mathrm{T}}\right]^{\mathrm{T}} \tag{3.74}$$

The aggregated error vector $\widetilde{\mathbf{c}}_q(\mathbf{k})$ is then defined as:

$$\widetilde{\mathbf{c}}_q(\mathbf{k}) \triangleq \widetilde{\bar{\mathbf{q}}} - \widetilde{\mathbf{f}}_q(\mathbf{k}), \quad \widetilde{\mathbf{c}}_q,\ \widetilde{\bar{\mathbf{q}}},\ \widetilde{\mathbf{f}}_q \in \mathbb{R}^{3N \times 1} \tag{3.75}$$

To optimize **k** we define an objective function M_q capturing the weighted error over the calibration data set:

$$M_q(\mathbf{k}) = \tfrac{1}{2N} \widetilde{\mathbf{c}}_q{}^{\mathrm{T}} \mathbf{W} \widetilde{\mathbf{c}}_q \tag{3.76}$$

Using the first-order Taylor series approximation of M_q and following a derivation as in section 3.5.1, we obtain **k** via nonlinear least squares using the aggregated Jacobian $\mathbf{J}_{c_q}$.

$$\mathbf{J}_{c_q} \triangleq \frac{\partial \widetilde{\mathbf{c}}_q}{\partial \mathbf{k}} = \left[\left(\frac{\partial \mathbf{c}_{q_1}}{\partial \mathbf{k}} \right)^{\mathrm{T}}, \ldots, \left(\frac{\partial \mathbf{c}_{q_N}}{\partial \mathbf{k}} \right)^{\mathrm{T}} \right]^{\mathrm{T}} \tag{3.77}$$

3.5.4 *Capturing the Augmented Parameter Vector $\widehat{\mathbf{k}}_T$ for a Multi-segment Continuum Robot*

For a n-segment continuum robot, let us use $\bar{\mathbf{x}}_{ij}$ and $\bar{\mathbf{R}}_{ij}$ to denote the position and orientation measured by a tracker attached to the end-effector of the i^{th} segment at the j^{th} robot configuration. Also, let us assume that all the measurements are obtained in the base of the 1^{th} segment. Similarly, the symbols $\mathbf{x}_{ij}$ and $\mathbf{R}_{ij}$ will denote the modeled position and orientation using an extension of the single segment numerical direct kinematics (section 3.3.1) to a multi-segment case.

$$\mathbf{c}_{T_{ij}} \triangleq \left[(\bar{\mathbf{x}}_{ij} - \mathbf{x}_{ij})^{\mathrm{T}}, (\alpha_{e_{ij}} \hat{\mathbf{m}}_{e_{ij}})^{\mathrm{T}}\right]^{\mathrm{T}}, \quad \mathbf{c}_{T_{ij}} \in \mathbb{R}^6 \tag{3.78}$$

where $\alpha_{e_{ij}}$ and $\hat{\mathbf{m}}_{e_{ij}}$ are defined, in a similar manner as in Eq. (4.60) while using $\mathbf{R}_{e_{ij}} \triangleq \bar{\mathbf{R}}_{ij}\mathbf{R}_{ij}^{\mathrm{T}}$ instead of $\mathbf{R}_{e_j}$.

An augmented task space error vector $\widehat{\mathbf{c}}_{T_j}$ is defined to capture the errors between the measurements and modeled poses for all n segments of the j^{th} robot configuration:

$$\widehat{\mathbf{c}}_{T_j} \triangleq \left[(\mathbf{c}_{T_{1j}})^{\mathrm{T}}, (\mathbf{c}_{T_{2j}})^{\mathrm{T}}, \ldots, (\mathbf{c}_{T_{nj}})^{\mathrm{T}}\right]^{\mathrm{T}} \mathbf{c}_{T_j} \in \mathbb{R}^{6n} \tag{3.79}$$

Finally, an aggregated error $\widetilde{\mathbf{c}}_T$ for all calibration poses is defined as:

$$\widetilde{\mathbf{c}}_T = \left[(\widehat{\mathbf{c}}_{T_1})^{\mathrm{T}}, (\widehat{\mathbf{c}}_{T_2})^{\mathrm{T}}, \ldots, (\widehat{\mathbf{c}}_{T_N})^{\mathrm{T}}\right]^{\mathrm{T}} \qquad \widetilde{\mathbf{c}}_T \in \mathbb{R}^{6nN\times 1} \tag{3.80}$$

Recall from Eq. (3.42), the error propagation of a multi-segment continuum robot can simply be expressed using the augmented identification Jacobian matrix, $\widehat{\mathbf{J}}_{\mathbf{k}_T}$, as:

$$\delta\widehat{\boldsymbol{\xi}} = \widehat{\mathbf{J}}_{\mathbf{k}_T}\delta\widehat{\mathbf{k}}_T, \qquad \delta\widehat{\mathbf{k}}_T = \left[(\delta\mathbf{k}_{T_1})^{\mathrm{T}}, \ldots, (\delta\mathbf{k}_{T_n})^{\mathrm{T}}\right]^{\mathrm{T}} \tag{3.81}$$

Applying this error propagation on the error vector $\delta\mathbf{c}_{T_j} = -\delta\boldsymbol{\xi}$ for one robot configuration,

we obtain:

$$\delta \mathbf{c}_{T_j} = -\widehat{\mathbf{J}}_{\mathbf{k}_T} \delta \widehat{\mathbf{k}}_T \tag{3.82}$$

For a given j^{th} configuration $\boldsymbol{\psi}_{1j}, \ldots, \boldsymbol{\psi}_{nj}$ we define an augmented configuration vector $\boldsymbol{\Psi}_j$ defined as:

$$\boldsymbol{\Psi}_j = \left[(\boldsymbol{\psi}_{1j})^{\mathrm{T}}, (\boldsymbol{\psi}_{2j})^{\mathrm{T}}, \ldots, (\boldsymbol{\psi}_{nj})^{\mathrm{T}} \right]^{\mathrm{T}}, \quad \boldsymbol{\Psi}_j \in \mathbb{R}^{2n \times 1} \tag{3.83}$$

To capture the error of the augmented parameters for all calibration poses we define the aggregated identification Jacobian $\mathbf{J}_{c_T} \in \mathbb{R}^{6nN \times 8n}$:

$$\mathbf{J}_{c_T} \triangleq \frac{\partial \widetilde{\mathbf{c}}_T}{\partial \widehat{\mathbf{k}}_T} = -\left[\left(\widehat{\mathbf{J}}_{\mathbf{k}_T}(\bar{\boldsymbol{\Psi}}_1) \right)^{\mathrm{T}}, \ldots, \left(\widehat{\mathbf{J}}_{\mathbf{k}_T}(\bar{\boldsymbol{\Psi}}_N) \right)^{\mathrm{T}} \right]^{\mathrm{T}} \tag{3.84}$$

$\widehat{\mathbf{k}}_T$ can then be estimated using nonlinear least squares.

3.5.5 Parameter Scaling

The numerical nonlinear least squares algorithm for estimating the calibration parameters relies on a numerical update step involving the inversion of the identification Jacobian (step 5 in Algorithm 3). Since the calibration parameters have different units they may have significantly different numerical ranges for the estimated variables, which can contribute to poor numerical conditioning. To overcome this problem, we adopt the *column normalizing* method [55][9].

For brevity, we will use the two placeholders $\widetilde{\mathbf{c}}_x$ and $\mathbf{k}_x$ to denote the various aggregated error and parameter vectors defined previously[10]. We also use $\breve{(\cdot)}$ to denote "scaled" terms. Parameter scaling is next introduced using an invertible weighting matrix $\mathbf{H}$. Following [40], we choose:

$$\mathbf{H} \triangleq \mathrm{diag}([h_1, h_2, \ldots, h_m]), \quad m = \mathrm{length}(\mathbf{k}_x) \tag{3.85}$$

[9]The column *extremal scaling value* method has some added advantages [82], but for simplicity we used the column normalization method

[10]For instance, $\widetilde{\mathbf{c}}$ may be replaced with $\widetilde{\mathbf{c}}_p$, with $\mathbf{k}_x$ being replaced with $\mathbf{k}_\kappa$.

The vector $\delta\mathbf{k}_x$ is scaled along with the corresponding identification Jacobian $\mathbf{J}_{c_x}$ such that:

$$\delta\mathbf{k}_x \rightarrow \mathbf{H}^{-1}\delta\mathbf{k}_x \triangleq \delta\breve{\mathbf{k}}_x, \quad \mathbf{J}_{c_x} \rightarrow \mathbf{J}_{c_x}\mathbf{H} \triangleq \breve{\mathbf{J}}_{c_x} \tag{3.86}$$

The numerical update using the scaling matrix becomes:

$$\delta\widetilde{\mathbf{c}}_x = \breve{\mathbf{J}}_{c_x}\delta\breve{\mathbf{k}}_x \quad \rightarrow \quad \delta\breve{\mathbf{k}}_x = \breve{\mathbf{J}}^{+}_{c_x}\delta\widetilde{\mathbf{c}}_x \tag{3.87}$$

The matrix $\mathbf{H}$ is constructed using:

$$h_i = \begin{cases} \|\mathbf{J}^i_{c_x}\|^{-1} & \text{if } \|\mathbf{J}^i_{c_x}\| \neq 0 \quad (3.88a) \\ 1 & \text{if } \|\mathbf{J}^i_{c_x}\| = 0 \quad (3.88b) \end{cases}$$

where $\mathbf{J}^i_{c_x}$ is the i^{th} column of the aggregated $\mathbf{J}_{c_x}$.

3.6 Simulations of Error Propagation on Geometric Parameters

Before we investigate parameter estimation (kinematic model calibration) via simulations, in this section, we present a means of using error propagation to provide insights on sensitivity of different parameters. We evaluate the effect of calibration errors in three cases. In case I (3.6.1) we evaluate the effect of twist about the primary backbone on the accuracy of the robot if this twist is not accounted for. In case II (3.6.2) we evaluate the effect of errors in (k_θ, k_δ) on the accuracy of the robot. Finally, in case III (3.6.3) we evaluate the effect of home position errors in $\mathbf{q}_h$ on the accuracy of the robot. In this section we used a single-segment robot with $L = 60\,mm$, $r = 4\,mm$, $\gamma_{e0} = 10^\circ$, $\mathbf{q}_h = [1, -2, 3]\ mm$.

3.6.1 *Effect of Twist about the Primary Backbone*

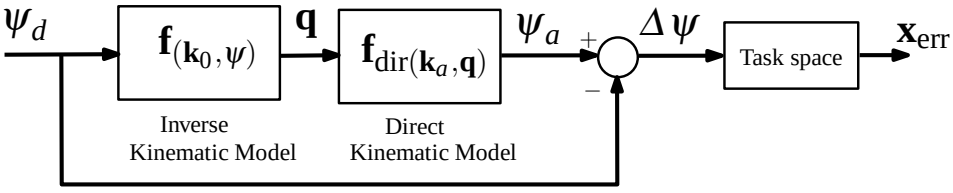
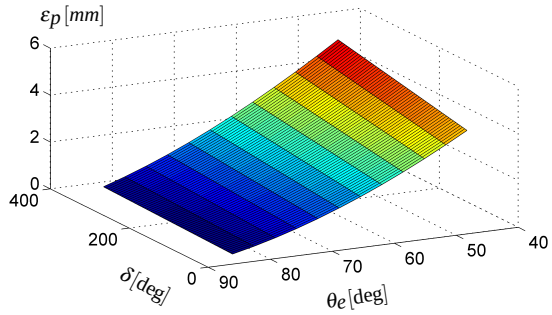


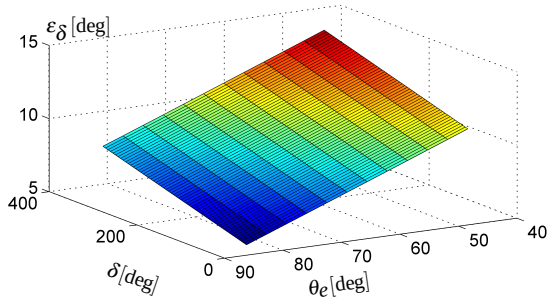
Figure 3.5: Simulation flow chart

Figure 3.5 shows the flowchart of the simulation for evaluating the effect of an unknown twist angle γ_e on the task and configuration space errors. A joint actuation $\mathbf{q}$ is planned based on given desired configuration ψ_d and an ideal model $\mathbf{k}_0$ assuming non-twisted and circular bending shape. This joint actuation is fed into the forward kinematics model using $\mathbf{k}_{actual}$ which includes twist errors but satisfies the circular bending assumption. The error $\Delta\psi$ is then propagated to the robot task space kinematics to evaluate the task space error.

We assumed in this simulation $\gamma_{e\max} = 30^\circ$, $k_\theta = 0.2327$, $k_\delta = 0.0185$, $a = -0.05/L$, $b = -0.8/L$. The errors between desired and actual robot pose are reported in configuration space and task space as shown in figure 3.6. For space limitations and because the maximal error in θ_e was less than 0.5° we did not include a separate for $\Delta\theta_e$. Figure 3.6(a) shows that the position error can be significant (almost 10% of L). Figure 3.6(b) shows that the error in the bending plane direction can be as high as 14.3°.



(a) end-disk position error



(b) end-disk rotation error

Figure 3.6: End-disk pose errors between twist-assumed model and non-twist-assumed model

3.6.2 *Effect of Twist Coefficient Gain Errors*

To further evaluate the sensitivity of the robot to twist we assume that the calibration parameters k_θ and k_δ are contaminated with 10% error. The nominal k_θ and k_δ were the same as in case 1. The jacobian $\mathbf{J}_k$ was used to calculate the propagation of errors using $\Delta\mathbf{q} = \mathbf{J}_k\Delta\mathbf{k}$. The configuration space error $\Delta\psi$ can be bounded using the singular value decomposition of $\mathbf{J}_{q\psi}$:

$$\frac{1}{\sigma_1} \leq \left\| \frac{\Delta\psi}{\Delta\mathbf{q}} \right\| \leq \frac{1}{\sigma_m},\ [\sigma_1, \sigma_2, \ldots, \sigma_m] = \mathrm{SVD}(\mathbf{J}_{\mathbf{q}\psi}) \tag{3.89}$$

Figure 3.7 shows the upper and lower bound for $\left\|\Delta\psi\right\|$.

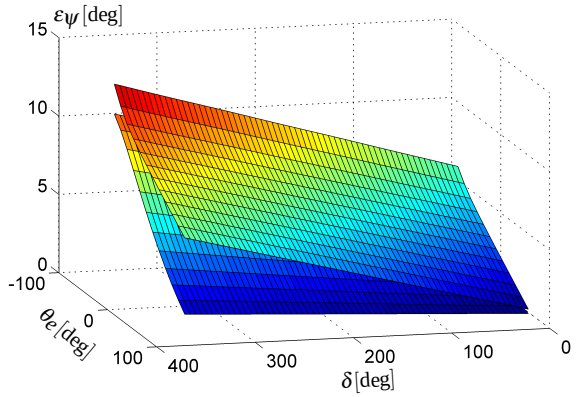


Figure 3.7: Configuration space error bounds calculated using error propagation - error due to $\Delta k_\theta, \Delta k_\delta$.

3.6.3 *Effect of Error in Joint Values at Home Position*

We also evaluated the effect of $\Delta\mathbf{q}_h$ on the robot configuration. Figure 3.8 shows the results for the upper and lower bounds of the error. We assumed an error of $\Delta\mathbf{q}_h = [0.1, 0.2, -0.3]^T$ [mm], and estimated $\Delta\mathbf{q} = \mathbf{J}_k\Delta\mathbf{q}_h$ and used equation 3.89 to bound the error.

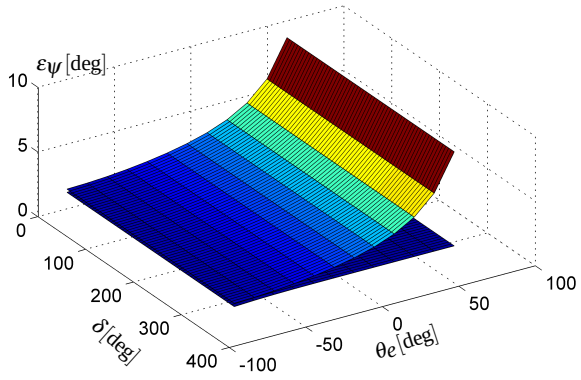


Figure 3.8: Configuration space error bounds calculated using error propagation - error due to $\Delta\mathbf{q}_h$.

3.7 Simulations of Parameter Estimation

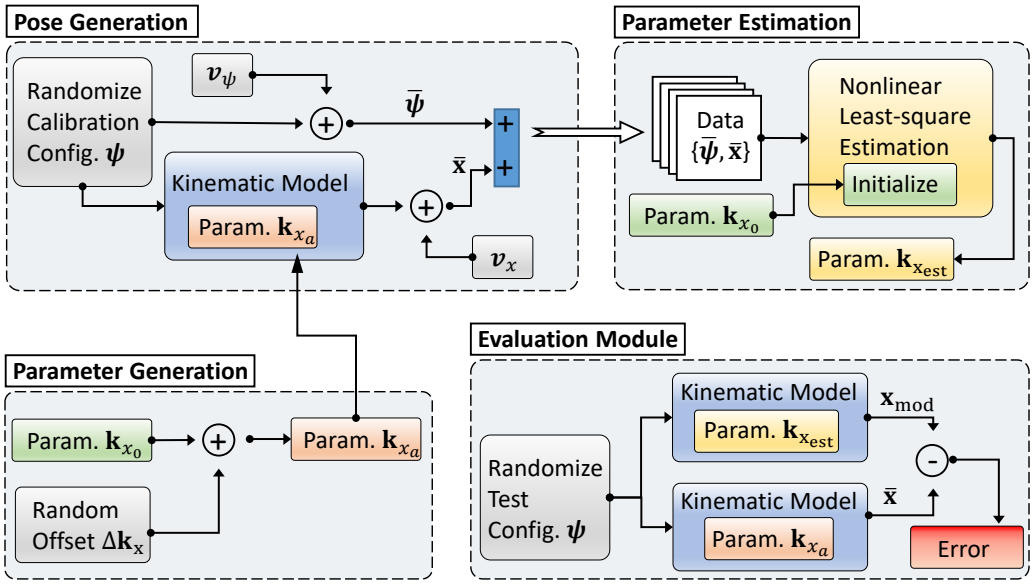


Figure 3.9: Work flow chart of all calibration simulation studies

Four conditions were used in simulations to investigate the sensitivity and feasibility in identifying $\mathbf{k}_\kappa$, $\mathbf{k}_\gamma$, $\mathbf{k}$ and $\widehat{\mathbf{k}}_T$, following the formulations as in sections 3.5.1-3.5.4, respectively. All simulations follow a similar work-flow as in Fig. 3.9 where an example of calibration using configuration to task space kinematic model ($\mathbf{k}_\kappa$ calibration) is illustrated.

In the following discussion, we will use $\mathbf{k}_x$ as a placeholder for the estimates of either one of $\mathbf{k}_\kappa$, $\mathbf{k}_\gamma$, $\mathbf{k}$ and $\widehat{\mathbf{k}}_T$ and $\mathbf{k}_{x_a}$ as its actual value. Referring to Fig. 3.9, the *parameter*

generation module generates a random offset $\Delta\mathbf{k}_x$ and applies it on a pre-defined *nominal* parameter vector $\mathbf{k}_{x_0}$, resulting in the *actual* parameter vector $\mathbf{k}_{x_a}$. This actual parameter $\mathbf{k}_{x_a}$ is then used in the *pose generation* module to calculate simulated input data needed for calibration algorithm, e.g. the position and orientation of the robot tip, the configuration space vector, or the joint space vector. During pose generation, the workspace is sampled uniformly, and sensor noises are simulated under different conditions as in Table 3.2 (denoted as $\boldsymbol{\nu}_x$ in Fig. 3.9).

The *parameter estimation* module uses nonlinear least-square methods to obtain the parameter estimates $\mathbf{k}_{x_{\text{est}}}$. Finally, in the *evaluation* module, the estimated parameter vector is compared to the actual parameter vector via the same computational kinematic models, where errors in both parameter values and in predicted poses are captured.

All simulation experiments in the following sections used nominal parameters: $L_0 = 60\,\text{mm}$, $r = 4\,\text{mm}$, representing approximately the length and the radius of a continuum robot segment. We also initialized[11] the modal coefficients in Eq. (3.5) with the following values:

$$\mathbf{a}_0 = [-0.05/L_0, -0.01/L_0]^{\mathrm{T}}, \qquad \mathbf{b}_0 = [-0.5/L_0, -0.15/L_0]^{\mathrm{T}} \tag{3.90}$$

In addition, ten simulation trials were conducted for each condition listed in Table 3.2, in which randomized zero mean Gaussian measurement noise was added with different noise variance in position and orientation. Table 3.2 lists the root mean squared position and orientation measurement noise assumed in each condition. In addition, we assumed $\pm 25\%$ random offsets on the nominal values of $\mathbf{a}_0, \mathbf{b}_0$ to generate actual parameter values. For conditions assuming a second sensor placed at approximately $L_0/2$ arc length, we assumed a uniform random noise with the range of $\pm 25\% L_0/2$. The simulation measurements were uniformly distributed throughout the workspace with 10 values of t selected from 0-1 and 15 values of δ selected from 0° to 360°, resulting in 150 calibration poses in total.

[11] We will henceforth use subscript zero to denote initialization of a variable that is updated during the estimation process

3.7.1 Estimation of the Shape Characteristic Parameter $\mathbf{k}_\kappa$

Condition	Simulation Conditions				Simulation Results	
	Sensor at End-disk	Sensor in Middle	Position Noise [mm]	Rotation Noise [degree]	Avg. Pos. RMSE [mm]	Avg. Rot. RMSE [degree]
1	Yes	No	0.5	1°	**0.67**	**0.79°**
2	Yes	Yes	0.5	1°	**0.74**	**0.82°**
3	Yes	No	2	3°	**2.61**	**2.74°**
4	Yes	Yes	2	3°	**3.05**	**2.86°**

Table 3.2: Table of different conditions used in shape parameter estimation simulations.

Parameter values obtained from the parameter estimation module were evaluated for each simulation trial: position root mean square errors (RMSEs) and bending angle RMSEs were captured between the modeled poses and simulated pose measurements over all samples, and the result is reported in Fig. 3.10. By comparing condition 1 with condition 2, and by comparing condition 3 with condition 4, we observe that a redundant sensor in addition to the one at the robot tip does not significantly help. Considering the different noises added for different conditions, we also observe that the residual errors are similar to the noise level when the algorithm converges.

3.7.2 Estimation of the Twist Characteristic Parameter $\mathbf{k}_\gamma$

Simulations of estimating $\mathbf{k}_\gamma$ based on prior estimates of $\mathbf{k}_\kappa$ facilitate investigating the sensitivity and accuracy in identifying $\mathbf{k}_\gamma$ independently. A more efficient alternative is to estimate $\mathbf{k}_\kappa$ and $\mathbf{k}_\gamma$ simultaneously, even for a multi-segment robot, as illustrated later in section 3.7.4.

Using the estimated $\mathbf{k}_\kappa$ from trial 1 of simulation condition 1, five simulation trials were done following Fig. 3.9 workflow. The nominal parameter values $\mathbf{k}_{\gamma_0}$ were: $\gamma_{e_0} = 10^\circ$, $k_\theta = 1$, $k_\delta = 0.1$, on which $\pm 20\%$ randomized offsets were applied to generate actual parameter $\mathbf{k}_{\gamma_a}$. The noise and sensor arrangement were consistent with condition 1 in Table 3.2.

Figure 3.11 shows the results of the five representative simulation trials out of ten

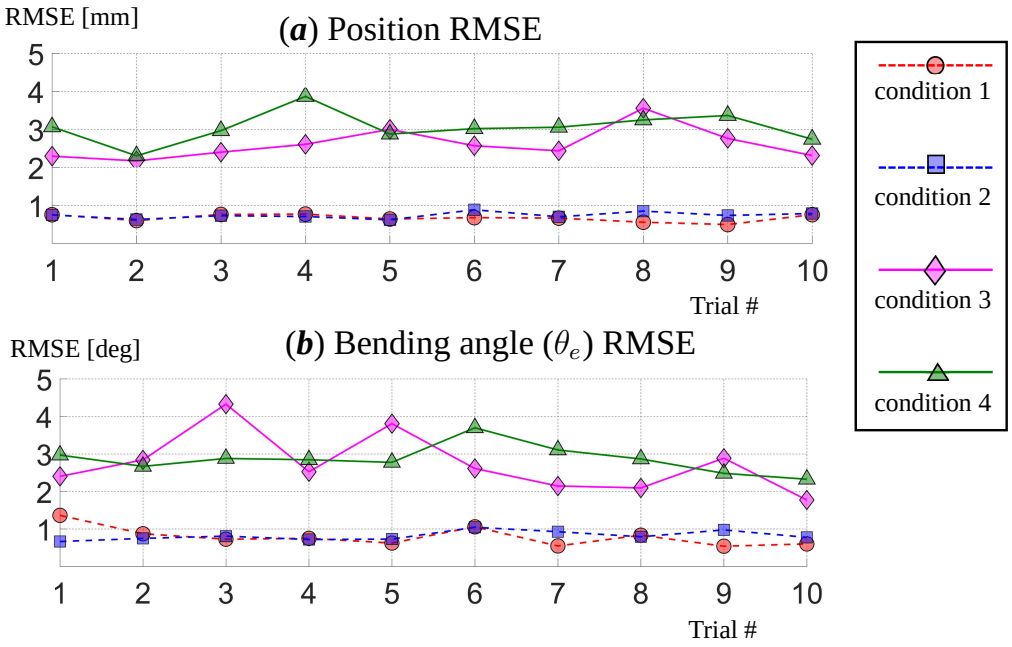


Figure 3.10: The RMSE of position and bending angle θ_e error after calibration of $\mathbf{k}_\kappa$.

conducted. Figure 3.11 (a) shows the rotation RMSEs for each iteration in the nonlinear least-square estimation. Figure 3.11 (b) shows the percentage error of the estimates of the parameters. The figure shows rapid convergence within less than 10 iterations and a significant reduction in orientation error. After calibration, the average RMSE for orientation was 1.0° and the averaged estimated errors of parameters γ_{e0}, k_θ and k_δ were 1.22%, 0.27% and 0.44% respectively.

3.7.3 Estimation of the Overall Characteristic Vector **k**

An alternative approach to estimating $\mathbf{k}_\kappa$ followed by estimating $\mathbf{k}_\gamma$ is to directly estimate the overall calibration characteristic $\mathbf{k}$ as defined in Eq. (3.53) using the method described in 3.5.3. This simulation study was carried our to elucidate the advantages or pitfall of this approach.

Figure 3.12 shows that for a large number (approximately 200) of calibration poses the NLS algorithm converges to a low $\tilde{\mathbf{c}}_q$ (joint position norm error defined in Eq. (3.75)) of 0.05 mm matching the measurement noise. However, the estimated overall characteristic $\mathbf{k}$

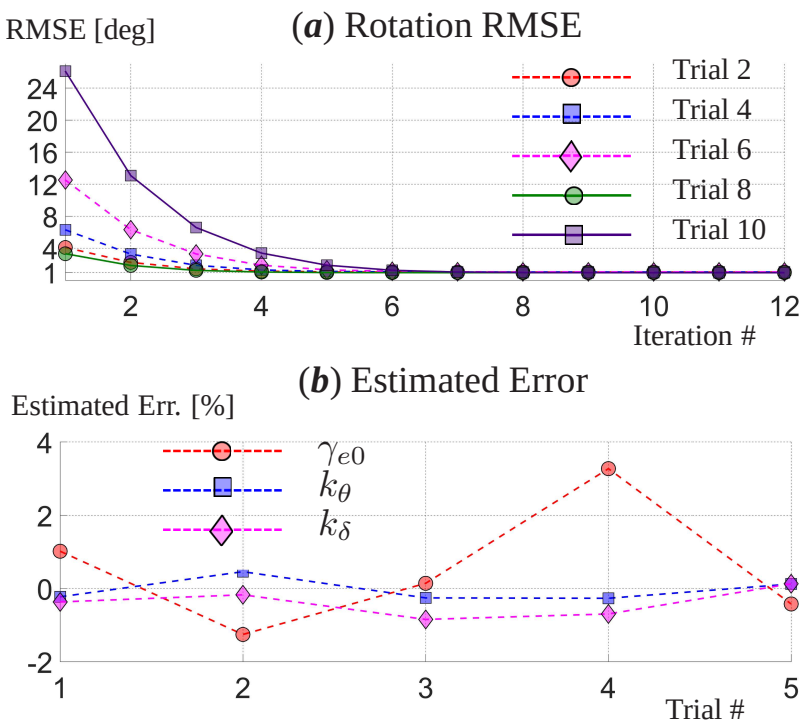


Figure 3.11: Simulation results for twist characterization: rotational residual of different iterations during nonlinear least-squared estimation and the resulted estimation errors. Results show 5 simulation trials for reducing figure clutter.

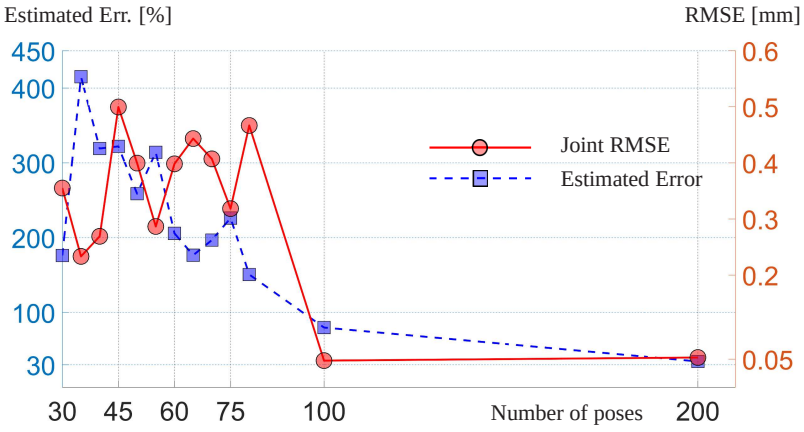
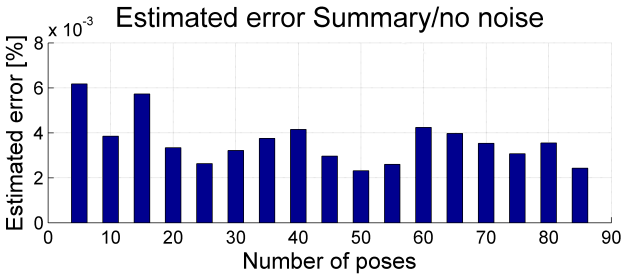


Figure 3.12: Calibration error for different numbers of calibration poses: (right axis) joint position RMSE in mm; (left axis) estimated percentage error in overall characteristic **k**

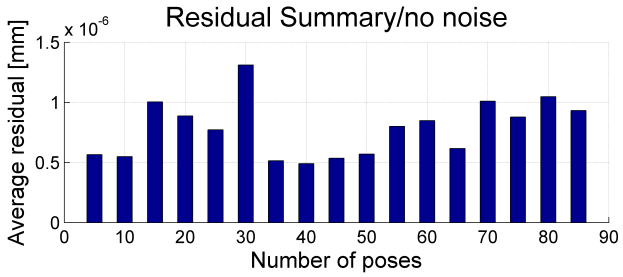
exhibits poor calibration results averaging 35% norm error even if one uses 200 calibration poses. The estimates of the curvature profile parameters were poor and the observability indices calculated as in [67] and the noise amplification index was very low **k**.

The simulations show that the direct relationship between measurements in configuration space and joint space is not sensitive enough to capture the kinematic parameters. This difficulty stems from the use of parallel routing of backbones, which causes the length of the i^{th} secondary backbone to be independent from the bending shape [37, 88].

To verify the calibration algorithm computation, we investigate the calibration using simulations assuming no noise added. The results from simulations without noise are shown in Figure 3.13. Figure 3.14 shows the observability indices that are calculated following [67]. From the results we can see that in ideal data cases, the number of poses does not affect either estimated error or final residual and a small number of poses are sufficient to achieve a good estimation. However the observability indices O_2 and O_4 are very poor, which corresponds to noise amplification.

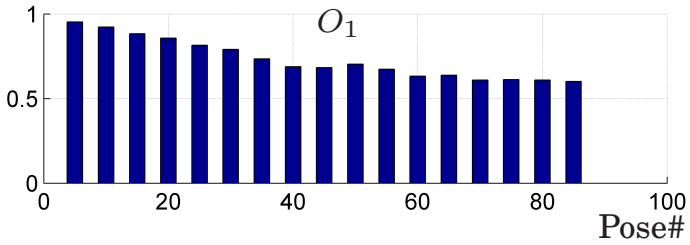


(a) estimated errors

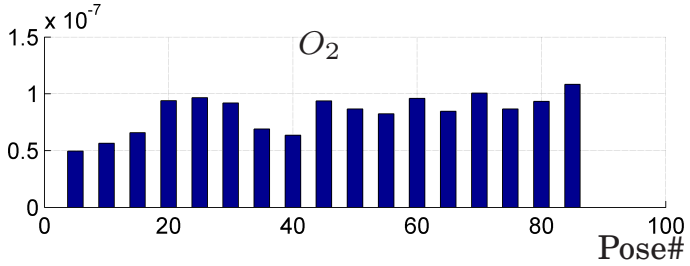


(b) residual errors

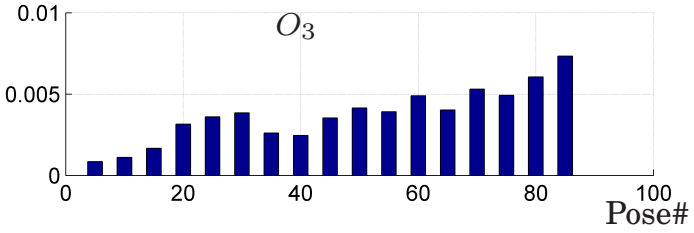
Figure 3.13: Estimated error and residual in simulations without noise



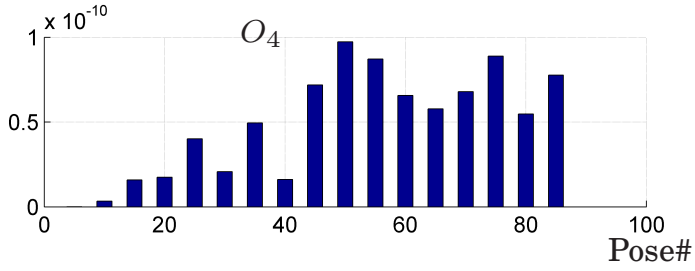
(a) O_1



(b) O_2



(c) O_3



(d) O_4

Figure 3.14: Observability indices in simulations without noise

3.7.4 Estimation of the Augmented $\widehat{\mathbf{k}}_T$ of a Multi-segment Robot

A simulation case study of a 3-segment continuum robot was used to verify our formulation in section3.5.4. In total, 10 simulation trials were carried out under condition 1 in Table 3.2.

In each simulation, each of the 3 segments was assigned with different shape and twist characteristic parameters as their "actual" parameters, where random offsets up to 25% were applied on the same nominal values as before. Simulated pose measurements, $[\bar{\mathbf{x}}, \bar{\mathbf{R}}]$, along with simulated configuration space measurements, $\bar{\mathbf{\Psi}}$, were generated as calibration input based on the "actual" kinematic parameters. In each simulation, the robot was simulated reaching 150 configurations, resulting in 450 end-disk pose measurements of all segments and 150 configuration space variable measurements.

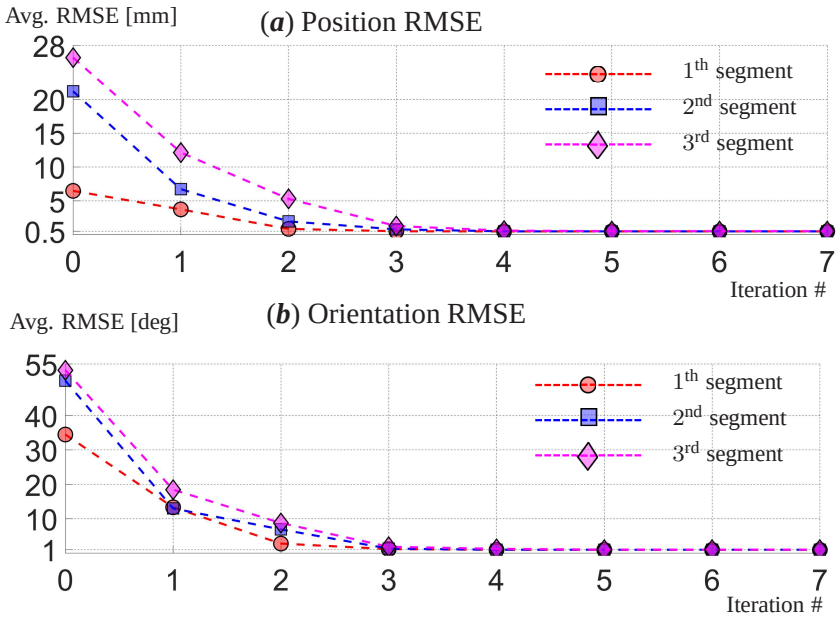


Figure 3.15: Simulation results of multi-segment robots: averaged position RMSE and orientation RMSE of different iterations during nonlinear least-squared estimation.

The augmented parameter vector, $\widehat{k_T}$, was estimated using the iterative nonlinear least-squares method as described in 3.5.4. For an iteration in one estimation, root mean square errors (RMSEs) were captured between the measured and the modeled pose of each segment end-disk. Positional and rotational RMSEs were reported for each segment individually because the segments are serially stacked and the error accumulates. The averaged positional and rotational RMSEs over the 10 simulation trials are reported in for all 3 segments in Fig. 3.15 . The average RMSEs are reported for each iteration. The results show a rapid convergence within less than 5 iterations to levels commensurate with the accumulated

measurement noise. The averaged RMSE over all segments after calibration was 0.49 mm in position and 0.97° in orientation.

3.8 Experimental Validation

3.8.1 Data Collection and Pre-processing for Calibration

In addition to simulation case studies, we also used the Insertable Robotic Effector Platform (IREP) [6, 22] as an experimental evaluation platform. As shown in Fig. 3.16, 6-DoF electromagnetic (EM) sensors (NDI Ascension DriveBAY™) were attached to the base, the 1th segment, and the 2nd segment (end-effector), respectively. In each experiment, a continuum robot segment was controlled to bend (or straighten) in one direction. The poses were recorded at a frequency of 125 Hz.

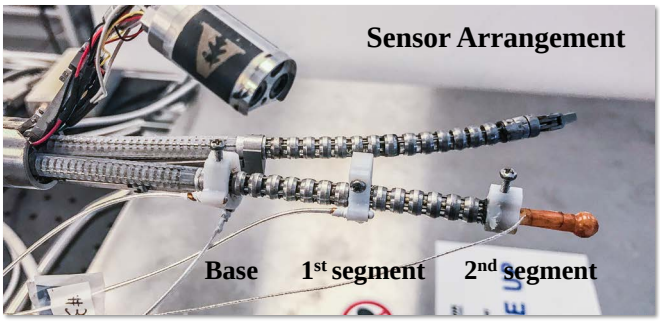


Figure 3.16: EM tracker experimental setup for data collection

To obtain the kinematic parameters, as in section 3.5 and 3.7, measurements of the configuration space ψ and the in-plane position and bending angle are needed. These measurements were obtained by processing the raw position and orientation tracked data from EM trackers as shown in Fig. 3.17: in (a) each pose was measured as the frame of the end-effector relative to the robot base using four EM trackers; in (b) the pose measurements of one bending experiment were used to generate the bending plane; and by projecting the poses onto the bending plane as in (c) we obtained the in-plane positions and bending angles, as shown in (d).

The bending plane normal was obtained as the singular vector associated with the

minimal singular value of the matrix containing the $\hat{\mathbf{z}}$ axes of all measured poses. Finding the maximum and minimum in-plane bending angles of one experiment, we obtained the value of the shape interpolation variable t as in Eq. (3.57).

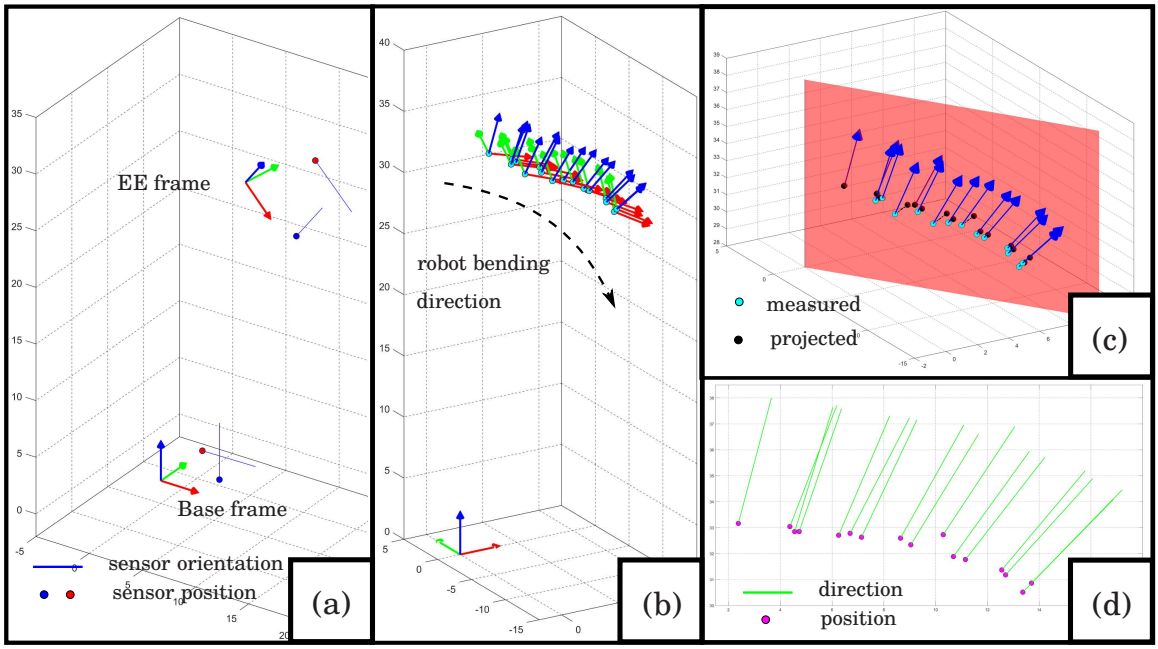


Figure 3.17: Illustration of processing the robot pose measurements for calibration use.

3.8.2 Shape Characterization Results

Following section 3.5.1 and using the prepared data pairs $\{\bar{\mathbf{x}}_p, \bar{t}\}$, the shape kinematic parameter estimation was done for multiple bending directions. Nonlinear least-squares estimation was used to capture the parameters, starting with an initial guess of the shape parameters - the length being the distance when the robot was least bent and the curvature coefficients representing circular shapes.

To evaluate the performance of the proposed shape calibration method, a comparison baseline was chosen as: a highly simplified calibration method assuming circular bending shape. The circular assumption simplifies Eq. (3.10) to Eq. (C.6), and the shape calibration

becomes a linear least-squared problem with only one parameter - length L.

$$\mathbf{p}_{xz} = [p_x, p_z]^{\mathrm{T}} = [1 - \sin\theta_e,\ \cos\theta_e]^{\mathrm{T}} \frac{L}{\theta_0 - \theta_e} \tag{3.91}$$

The results of the shape calibration of one bending direction is illustrated in Fig. 3.18, where the circular-assumed method is compared to the proposed method.

In total, experiments were conducted for 15 bending directions, at an incremental δ angle of 5.7° to cover $\delta \in [5°, 85°]$. In each bending group, 20 poses were collected for calibration use. Using this calibration data, we obtained the calibrated shape parameters as: $L = 32.13\,\text{mm}$, $a_0 = -1.07\,°/\text{mm}$, $a_1 = 0.04\,°/\text{mm}^2$, $b_0 = -2.83\,°/\text{mm}$, $b_1 = 0.07\,°/\text{mm}^2$, in which the non-circular coefficients (a_1 and b_1) contributed bending angle offsets at the tip to be 18.94° and 33.90° when the robot was least bent and was most bent respectively.

Position and bending orientation RMSE of each experiment were captured between the modeled and the measured pose. Table 3.3 presents the RMSEs of the proposed method (*Proposed*) and the circular-assumed method (*Circ. Calib.*).

Table 3.3: The RMSE comparison of the shape characterization experiments between the proposed calibration method and the circular-assumed calibration method: in-plane bending positions and orientations

Bending Direction δ =		5°	11°	16	22°	28°	34°	39°	45°
Position RMSE [mm]	Circ. Calib.	6.22	6.52	6.81	7.52	6.53	6.85	5.75	6.10
	Proposed	1.35	1.13	1.17	1.00	0.98	0.95	0.91	1.07
Orientation RMSE [deg]	Circ. Calib.	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Proposed	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Bending Direction δ =		51°	56°	62°	68°	74°	79°	85°	Avg.
Position RMSE [mm]	Circ. Calib.	5.15	5.24	5.04	4.17	3.33	3.04	2.05	**5.36**
	Proposed	0.97	0.98	0.86	0.92	0.84	1.02	1.00	**1.01**
Orientation RMSE [deg]	Circ. Calib.	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Proposed	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

From Table 3.3 we observe: (i) the averaged position RMSE result shows that the proposed calibration method significantly improves the positional error over the circular-

assumed method, reducing error from 5.36 to 1.01 mm.

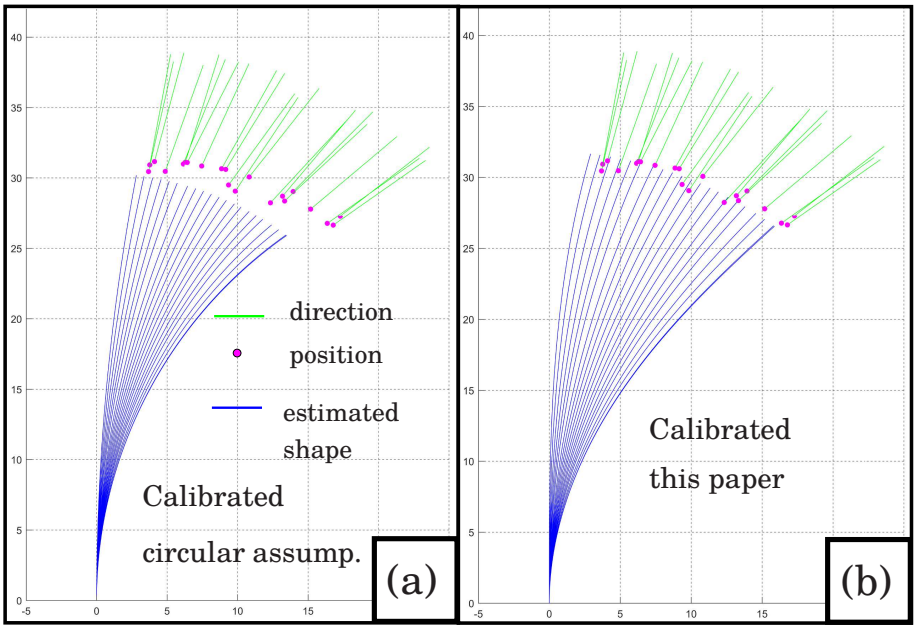


Figure 3.18: The result of one shape characterization experiment to illustrate: (a) bending shapes calibrated using circular assumption; (b) bending shapes calibrated using the proposed method.

To validate the shape characterization results intuitively, an HD camera was used to capture images of the robot bending. The ground truth shape of the robot bending was manually segmented as illustrated in Fig. 3.19. A series of continuous snapshots of the same bending experiment is shown in Fig. 3.20, overlayed with the modeled shape using the calibrated kinematic parameters.

3.8.3 Twist Characterization Results

Using the same experimental data collection and pre-processing procedure, the twist characterization algorithm was evaluated. In total, 450 robot pose measurements were collected while the robot was controlled to bend and to straighten in 15 different directions. For each of the bending group, a set of shape parameters was rapidly captured. A nonlinear least-square estimation, as discussed in 3.7.2, was performed to capture the twist kinematic parameters $\mathbf{k}_\gamma = [\gamma_{e_0}, k_\theta, k_\delta]^{\mathrm{T}}$. The estimation was initialized without twist i.e. $\mathbf{k}_{\gamma_0} =$

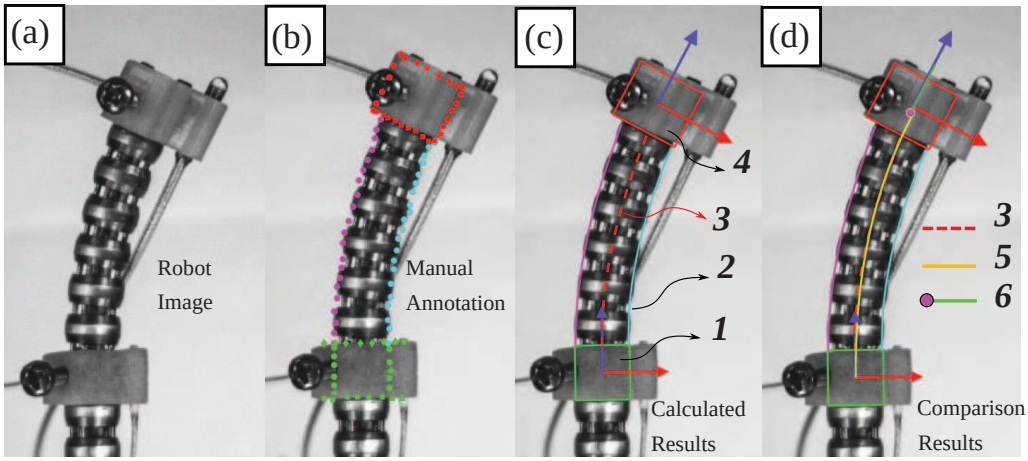


Figure 3.19: An image validation example of the shape characterization: (a) continuum robot segment; (b) manual annotation, including two boxes of the robot base and end-effector and two edges of the bending segment; (c) the calculated central backbone and end effector frame; (d) comparison of the processed image, overlayed with the characterized shape. ***1***, ***4*** - the fitted rectangles and frames of the base and end-effector using, [3] ***2***, ***3*** - the fitted 3rd order polynomial curves representing the segment edge and centerline respectively, ***5*** - the estimated shape from calibration, ***6*** - the EM tracker reading rewritten in the image frame.

$[0,0,0]^{\mathrm{T}}$.

Figure 3.21 shows that after 35 iterations, the root mean squared (RMS) rotational error converged from 12° to 2°. The estimation converged at a parameter result of $\gamma_{e_0} = -13.56°, k_\theta = 0.062, k_\delta = -0.025$. The result indicates that, for this particular robot assembly, a constant twist offset, γ_{e_0}, has more influence than bending angle or bending plane angle.

Fig. 3.22 illustrates the rotation errors between the modeled and the measured pose by plotting one sample in an experiment: the calibrated model and the model without twist are shown for visual comparison.

Root mean square errors were captured between modeled and measured orientations for each bending direction, and this evaluation was performed on the model including twist and model without twist. Results of this comparison are reported in Table 3.4.

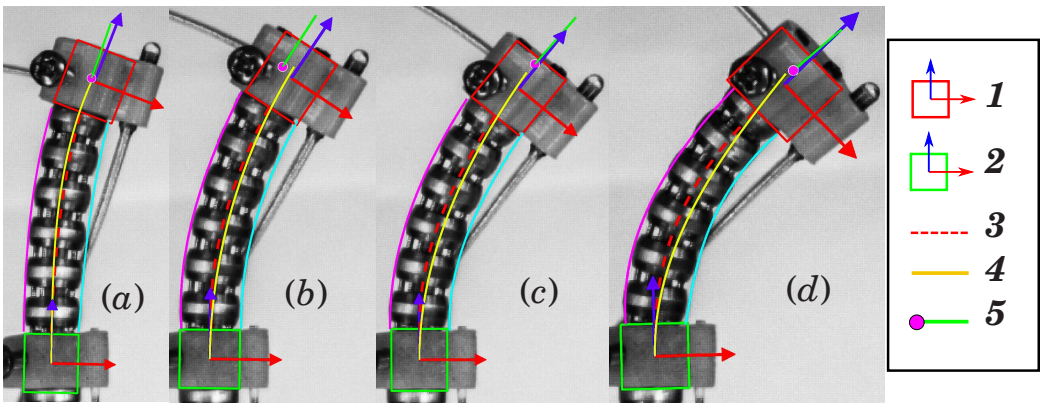


Figure 3.20: The validation snapshots of a continuous bending motion were captured while the estimated shapes calculated using the calibration algorithm were overlayed to compare. ***1***, ***2*** - the fitted rectangles and frames of the robot base and end-effector that are obtained from the annotated images, ***3*** - the fitted shape obtained from the annotated images, ***4*** - the estimated shape from calibration, ***5*** - the EM tracker reading rewritten in the image frame.

Table 3.4: The RMSE comparison of modeling the orientation between using the proposed method and using the non-twist model

Bending Direction $\delta =$		5°	11°	16	22°	28°	34°	39°	45°
Orientation RMSE [deg]	Non-twist	12.75	14.18	11.84	12.78	14.71	12.20	12.77	14.71
	Proposed	0.66	1.88	2.09	0.86	2.43	2.73	0.77	2.24
Bending Direction $\delta =$		51°	56°	62°	68°	74°	79°	85°	Avg.
Orientation RMSE [deg]	Non-twist	12.30	12.93	14.91	12.55	13.09	14.83	12.69	**13.32**
	Proposed	2.48	0.78	1.81	1.90	0.97	1.64	1.58	**1.79**

3.8.4 Integration and Evaluation of the Calibrated Model

To further investigate the feasibility of the proposed calibrated model, we next discuss the integration and evaluation of the calibrated model in a rectified real-time control framework [6]. The real-time control model operates *without* the use of magnetic tracker data, which was used to provide the shape interpolation variable t when formulating and solving the calibration problem.

For the evaluation, we used the IREP's twi-segment arms with the setup shown in Fig. 3.16. A real-time control code was implemented using Matlab® Simulink Real-Time™

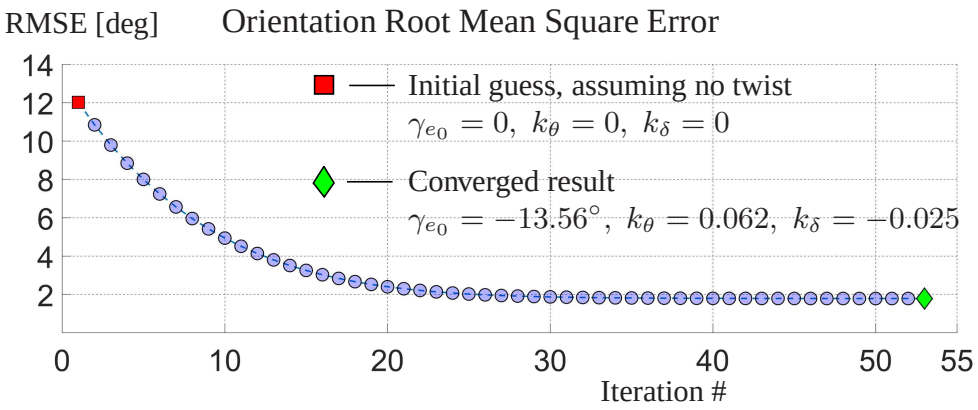


Figure 3.21: Orientation RMSE of iterations from the nonlinear least-square estimation to capture twist characterization using data from one bending experiment.

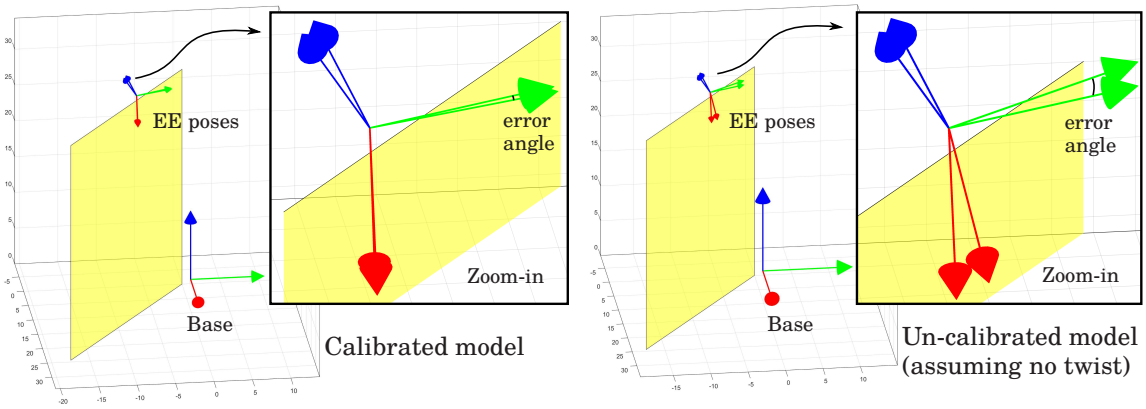


Figure 3.22: An illustration of rotational errors between the modeled poses and the measured poses.

at 1 kHz cycle frequency. An actuation compensation scheme based on [77], which compensates for actuation transmission motion losses due to extension of the actuation lines was used. The control code also used a resolved rates algorithm using an updated Jacobian $\mathbf{J}_{x\psi}$ based on the calibrated bending shape. The resolved rates algorithm provides a means for calculating the current ψ given a desired end-effector pose (position and orientation). Solving Eq. (3.3) for t for a given θ_e provides the input to our calibrated model control code without use of real-time tracking of the end effector pose for control purpose. Instead, the magnetic tracking was used for evaluating the accuracy of trajectory following. For trajectory following, a path plan was generated offline in configuration space and a resolved-

rates method was used in real-time control. The end effector was tracked using the magnetic tracker with position/orientation RMSE tracking of $1.5mm/0.5^{\circ}$. The real-time calibrated model is integrated with the actuation compensation mentioned above, as shown in Fig. 3.23.

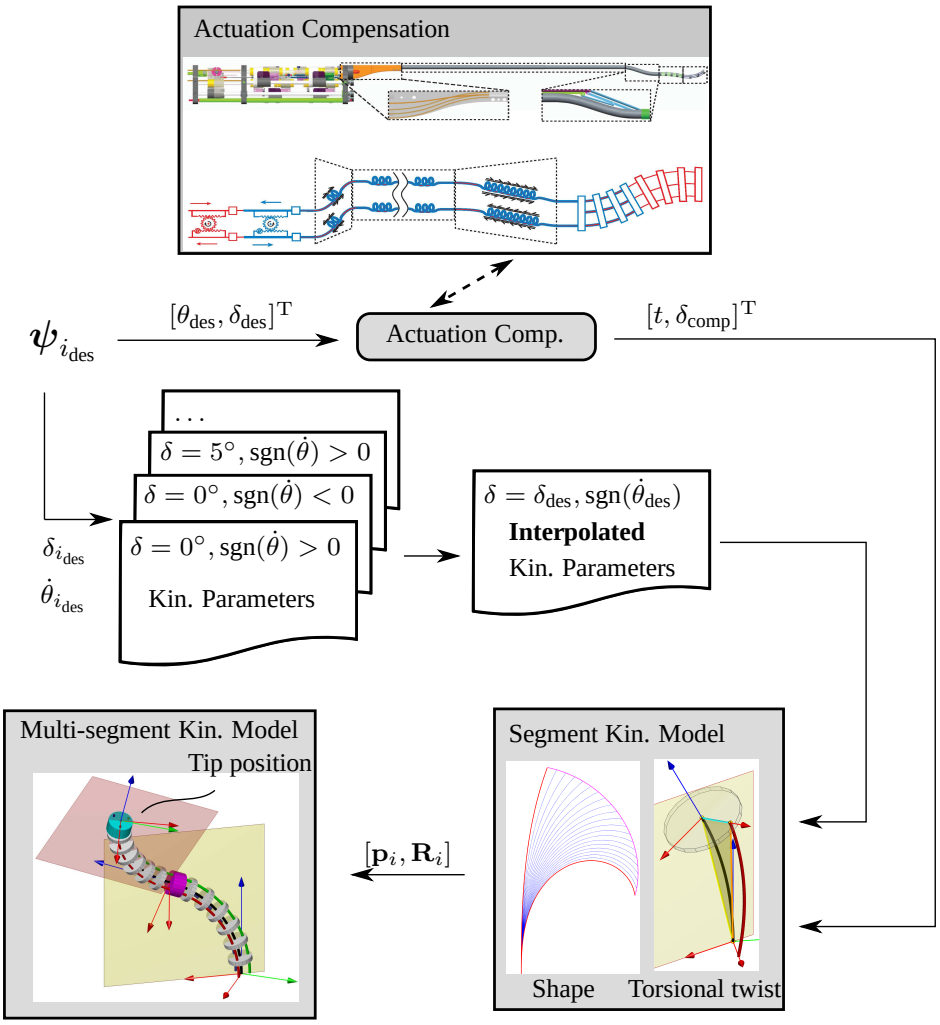


Figure 3.23: Real-time system integration solution of the calibrated kinematic model, combined with the actuation compensation.

To evaluate the performance of the calibrated model, the two-segment continuum arm of the IREP was commanded to follow a desired 3-dimensional 'V' trajectory as depicted by the dashed line in Fig. 3.24. The IREP arms include 7 degrees of freedom, but we used only four degrees of freedom since we wanted to valuate only the performance of the continuum portion of the robot in order to evaluate the effect of our calibration model. Therefore, we

specified a task of path following without orientation specification. When the calibrated non-circular bending model with twist was used, the robot tracked the v shape depicted in solid red line in Figure 3.24(a). The robot exhibited a position RMSE of 3.22mm. When a circular bending model was used based on measurement of segment lengths, the robot tracked the solid red line shown in Fig. 3.24(b) with an RMSE of 9.85mm. We also note that when a calibrated model using circular assumption to adjust the segment lengths was used the error was very high thereby we aborted evaluation. This was expected due to the constraint of matching the tip orientation to the measured tip angle while using circular bending model, which does not reflect the actual bending model (see for example the case of a single segment in Fig. 3.18(a)).

Though our method led to 305% improvement in tracking error, the tracking error RMSE was still high ($3.22mm$). This result can be explained by two factors: our calibration data includes the tracking RMSE of $1.5mm/0.5°$, which stack serially. In addition, our calibration model does not extend to calibrating our assumed model of actuation compensation based on [77].

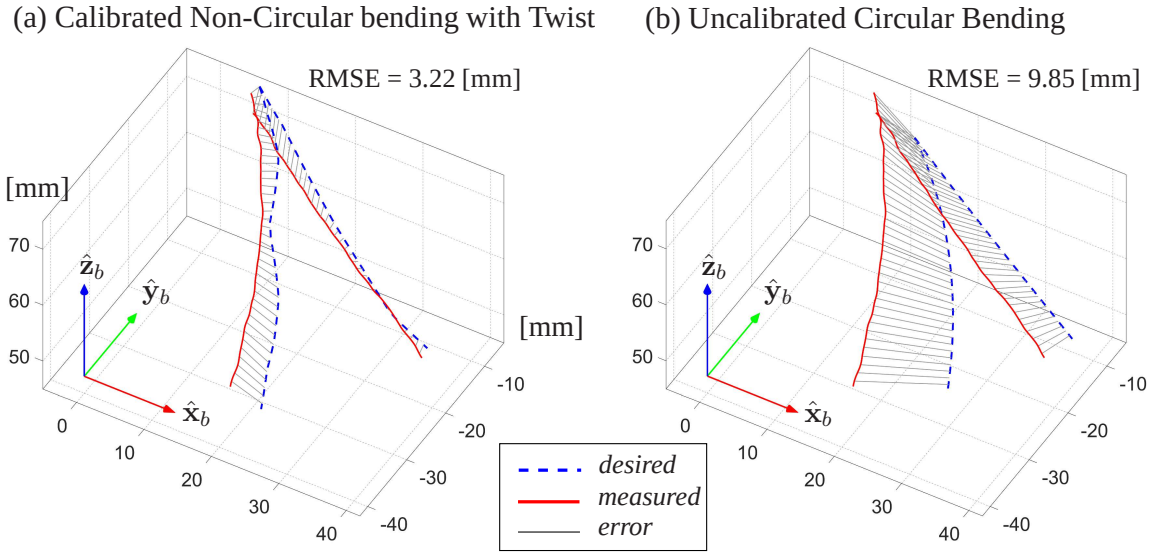


Figure 3.24: A "V" shape spatial trajectory following using (a) the proposed calibration and (b) an uncalibrated circular bending model based on measurement of segment length.

3.9 Conclusion

Current continuum robot modeling frameworks are not sufficiently accurate for autonomous tasks. To overcome these challenges, this paper presented a kinematic framework aiming at deriving calibration identification Jacobians of multi-backbone continuum robots. Two kinematic mappings, the configuration-to-task mapping, and the configuration-to-joint mapping, are derived for error prorogation and calibration. Errors in the configuration-to-task space mapping are captured through the use of bending shape homotopy, which introduces modal factors for calibrating the bending shape. Errors due to assembly are captured through modeling of twist along the central backbone. Errors in the joint-to-configuration mapping account also for errors in joint home values.

Several calibration alternatives are considered, including direct calibration of the resultant joint-to-task space mapping. It is shown that such direct calibration is not advisable due to its poor observability. Instead, a calibration of the configuration-to-task space mapping should be carried out independently. Also, it was shown that adding a larger number of sensors along the length of a continuum segment has little effect on improving the results of calibration - especially if one considers direct calibration of the joint-to-task space mapping. Simulation studies for multi-segment continuum robots validate our models and experimental results on a continuum robot for single port access surgery show a drastic improvement in the accuracy of the model.

In this work we assumed that a calibration of joint-space motion losses due to extension of the actuation lines has been carried out as a pre-requisite (e.g. such as the approach presented in [77]. While carrying out the calibration, we used the desired configuration space variables ψ_{des} to allow the calibration of configuration-to-joint space mapping. Residual errors of the joint-to-configuration space mapping affect the subsequent configuration-to-task space mapping and therefore one should first attempt to create as accurate a model as possible for compensating motion losses in these robots. A future expansion of our work would be the extension of the kinematic model to incorporate actuation line motion losses.

We however believe that such an approach should be investigated in detail since the two mappings (joint-to-configuration and configuration-to-task space) are not easily decoupled in the calibration process. Observability of the identification Jacobians for an extended model that also includes joint-to-configuration space mapping should be investigated.

Additional future improvements to overcome the limitations of this work include investigating calibration methods for continuum robots under loading. Such methods could benefit from the method presented here by producing multiple calibration maps for different loading, but such extension is not trivial and deserves further research. In addition, extension of the calibration method to include calibration of the actuation compensation models could possibly lead to additional improvements and will be the subject of future research.

We believe that the approach presented in this chapter can enable surgical continuum robots to accurately carry out semi-autonomous tasks involving small or no forceful interaction with the anatomy (e.g. laser ablation). In addition, this framework may be used in the future to initialize a kinetostatic model that takes into account deflections. To demonstrate the utility of our calibration approach, we have applied it to solving the calibration and modeling of several different flavors of continuum robots. In addition to first demonstrating it for the insertable robotic effectors platform (IREP) in [95, 97], we have applied our calibration approach to rectifying the kinematic model of a pneumatically-actuated continuum endoscope [31, 32] and to the modeling and calibration of a new type of continuum robots capable of micro-scale and macro-scale motion [21, 94].

3.10 Relevant Published Work

Published journal articles include:

J.1 **Wang, L.** and Simaan, N., "Geometric Calibration of Continuum Robots: Joint Space and Equilibrium Shape Deviations", *submitted to* ***IEEE Transactions on Robotics***, revision under review.

J.2 Roy, R., **Wang, L.**, Simaan, N., "Modeling and Online Estimation of Internal Friction and Actuation Compensation in Continuum Robots", ***IEEE Transactions on Mechatronics***,

Vol. 22, No. 2, pp. 990-920, 2017.

J.3 Garbin, N., **Wang, L.**, Chandler, J. H., Obstein, K. L., Simaan, N., and Valdastri, P., "Dual-Continuum Design Approach for Intuitive and Ultra Low-Cost Upper Gastrointestinal Endoscopy", ***IEEE Transactions on Biomedical Engineering***, 2018, under review.

J.4 Groom, K., **Wang, L.**, Simaan, N., Netterville, J., "Robot- Assisted Trans-nasal Laryngoplasty in Cadaveric Models: Quantifying Forces and Identifying Challenges", ***The Laryngoscope***, Vol. 125, No. 5, pp. 166-168, 2015.

J.5 Simaan, N., Bajo, A., Rieter, A., **Wang, L.** , Allen, P., Fowler, D. "Lessons Learned Using the Insertable Robotic Effector Platform (IREP) for Single Port Access Surgery", ***Journal on Robotic Surgery***, Vol. 7, pp. 235-240, 2013.

Published conference papers include:

C.1 **Wang, L.** and Simaan, N., "Investigation of Error Propagation in Multi-backbone Continuum Robots", proceedings of the 14th IFToMM (International Federation for the Promotion of Mechanism and Machine Science) Symposium on Advances in Robot Kinematics (ARK2014), Lenarčič, J. and Khatib, O. (Eds.), pp. 385-394, Ljubljana Slovenia, June 29-July 3, 2014.

C.2 Garbin, N., **Wang, L.**, Chandler, J. H., Obstein, K. L., Simaan, N., and Valdastri, P., "A Disposable Continuum Endoscope Using Piston-driven Parallel Bellow Actuator", In Medical Robotics (ISMR), 2018 International Symposium on, pp. 1-6. IEEE, 2018. *Best student paper award*.

C.3 Roy, R., **Wang, L.**, Simaan, N., "Investigation of effects of dynamics on intrinsic wrench sensing in continuum robots" IEEE International Conference on Robotics & Automation (ICRA')[12], pp. 2052 - 2059, Stockholm, Sweden, 2016.
(Finalist for best medical robotics paper).

C.4 Garbin, N., **Wang, L.**, Sohn, D., Obstein, K. L., Simaan, N., and Valdastri, P., "Design of a disposable endoscope with intrinsic pneumatic actuation", 29th Conference of the international Society for Medical Innovation and Technology (iSMIT), University of Torino, Italy, 2017.

C.5 Bajo, A., Goldman, R. E., **Wang, L.**, Fowler, D. and Simaan, N., "Integration and Preliminary Evaluation of an Insertable Robotic Effectors Platform for Single Port

[12]Although in some areas conference papers are considered non-peer reviewed publications, in my field all conference papers are peer reviewed. For example, ICRA is the largest robotics conference organized by IEEE society. The acceptance rate for ICRA was 40.25% in 2012, 38.5% in 2013, 48% in 2014, 41% in 2015, 35.1% in 2016.

Access Surgery", 2012 IEEE International Conference on Robotics and Automation, St. Paul, MI USA, pp. 3381-3387, 2012. *(finalist for best medical robotics paper)*.

Chapter 4

Simplified Kinematics of Continuum Robot Equilibrium Modulation via Moment Coupling Effects and Model Calibration

4.1 Concept of Continuum Robots with Equilibrium Modulation (CREM)

Current robotic manipulators for minimally invasive surgery (MIS) are capable of dexterous motion for surgical tasks requiring large workspace and position accuracy ranging from 0.5 to 1.5 mm. For example, the root mean square (RMS) localization accuracy of the da-Vinci Classic and da-Vinci S was evaluated experimentally as 1.02 mm and 1.05 mm respectively by Kwartowitz et al. [53, 54]. Despite recent increases in precision, current commercial surgical systems are unable to support micro-surgical precision (less than 0.1mm precision), and such precision can benefit micro-surgical tasks (e.g. micro-anastomosis and micro-vascular reconstruction [43, 90, 15, 60]).

This chapter is motivated by a need for increased motion resolution at a micro-surgical scale and during deep surgical access minimally invasive surgery. In addition, the paper is equally motivated by the potential benefits of a new class of surgical devices capable of multi-scale motion. Such devices promise to provide a large workspace for traversal of deep passageways and for gross surgical manipulation while offering micro-scale motion suitable for cellular-level surgery. We refer to devices capable of macro and micro-scale motion as Multi Scale Motion (MSM) devices. With the advent of new devices with integrated optical coherence tomography imaging and confocal endo-microscopy (e.g. [109, 113]), the use of MSM can allow future image-based biopsy with imaging resolutions at cellular size [29, 61]. Such devices can in the future support surgical decisions on continued tumor excision, which can minimize the need for repeat follow-up surgeries due to incomplete resection of tumors.

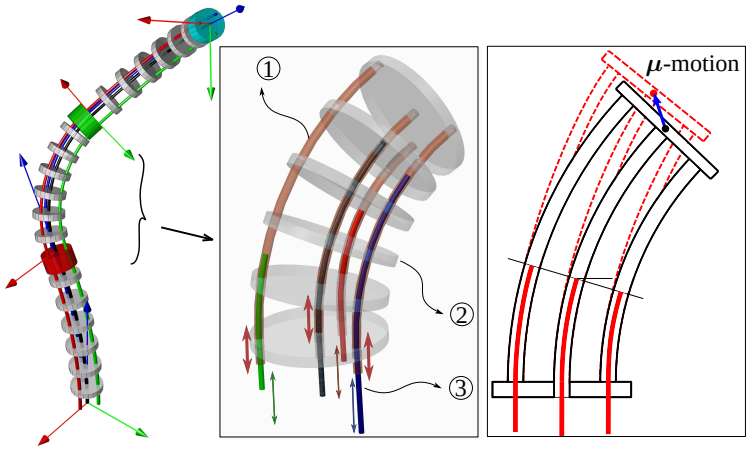


Figure 4.1: Continuum robots with equilibrium modulation (CREM): ① secondary tubular backbones, ② spacer disks, ③ equilibrium modulation backbones. The thick bi-directional arrows designate direct actuation of pushing and pulling on the secondary backbones while the thin bi-directional arrows designate the indirect actuation of the equilibrium modulation backbone insertion.

To achieve MSM capabilities, this paper adopts the new design concept for *Continuum Robot Equilibrium Modulation (CREM)*, which was first presented in [21]. CREM robots use a continuum structure that is primarily based on flexible elements to achieve large scale manipulation (i.e. robots without hinges and pin joints [74]). They also use fine adjustments to their static equilibrium pose in order to achieve micro-scale motion. The design concept for these robots is presented in Figure 4.1. This design is modified from a multi-backbone continuum robot presented in [86]. Each segment of a multi-segment continuum robot (MBCR) includes superelastic NiTi backbones. A single central backbone is surrounded by secondary backbones that are radially constrained by spacer disks and equidistantly distributed circumferentially around the central backbone. Macro motion of the robot tip is achieved by pushing and pulling on the secondary backbones (designated by the thick arrows in Fig. 4.1), which causes a deformation of the continuum robot body. We call this method of actuation *direct actuation* where the robot actuators directly change the length of the secondary backbones. In addition, CREM robots use *indirect actuation* whereby the equilibrium pose of the end effector is indirectly altered through a change of internal force distribution or by a change in material distribution altering cross sectional

stiffness. For example, by inserting superelastic Ni-Ti beams (henceforth referred to as the *Equilibrium Modulation Backbones* (EMBs)) inside the secondary backbones (see thin arrows in Fig. 4.1), the static equilibrium of the robot is altered (modulated) by minute amounts.

4.2 Related Work

Compared to prior designs, CREM robots possess a unique capability to allow MSM using a single design. Most prior works in the area of MSM rely on serial stacking of a macro-scale motion robot and a micro-scale robot, and such examples include Egeland's pioneering work [24] and followed by several other works such as [20, 39, 26, 1, 19, 49, 66]. Other researchers investigated a variety of actuation methods and mechanisms to achieve micro motion capabilities, including a Steward/Gough parallel robot driven by hydraulic micro-actuators [73], twisted wire actuators for a planar parallel robot [84], a micromanipulation tool using shape memory alloy [80], and a piezo-electrically actuated parallel platform [110]. Although these works achieved micro-scale motion resolution, they are not suitable for MSM in confined spaces, in which continuum robots in general have advantages.

Within the context of continuum robots, the most relevant modeling works are [104] where a solution framework based on constraints of geometric compatibility and static equilibrium was derived using elliptic integrals for multi-backbone continuum robots and [76, 79] where Cosserat rod theory was used for dynamics modeling of wire-actuated continuum robots. One could use these methods to model the tip micro motion, however, due to the formulation complexity and the solutions of equilibrium direct kinematics based on energy minimization or boundary value problem solution it is hard to obtain an updated differential kinematics model that accounts for the exact bending shape curvatures. Also, due to uncertainties in material properties and friction, using an exact modeling method does not add value since a model calibration step has to be carried out anyhow when

attempting to control a physical robot.

Another work is Li *et al.* [57], where the authors presented a constrained wire-driven flexible mechanism which used a constraint rod to selectively adjust its workspace by altering the length of its distal bending portion. The work showed that the constraint rod can change the workspace. The design however does not lend itself to easily allowing MSM and the work did not consider methods for achieving or modeling CREM.

Finally, our proposed design differs substantially from concentric tube robots [23, 100] in that the equilibrium modulation that creates the tip micro-motion is still governed by the strong geometric constraints employed by the parallel-backbone structure. Concentric tube continuum robots achieve their workspace through antagonistic bending of tube pairs and therefore they can theoretically be used for micro-scale equilibrium modulation. However to achieve micro-scale motion the designers are forced to use stiffness matched tube pairs with a very small difference in free curvature. The attainment of micro-scale motion by concentric tube robots therefore comes at the expense of sacrificing the macro-scale motion capabilities.

In contrast to the above-mentioned works, this paper takes a different approach. Instead of focusing on a high fidelity model, we present a simplified model that can be readily used to obtain the differential micro-motion kinematics Jacobian and is readily amenable to formulating a model calibration problem. This micro motion Jacobian is essential for control purpose, and an associated identification Jacobian is needed for calibration purposes. Therefore, the paper focuses on the derivation of the micro-motion kinematics and its associated identification Jacobian for calibration and error prorogation.

This work is built upon our previous work [21], in which we presented the concept of CREM and provided a visual measurement solution to observe micro-motion. The work in [21] lacked a modeling approach that can explain the experimental observations and that can be used for control and identification purposes. The contribution of this work is in presenting a simplified kinematic modeling framework that captures the micro-

motion achieved by the equilibrium modulation of continuum robots, and in developing a calibration approach to capture the model parameters. We put forth the concept of moment coupling effect as a simplified approach to describe the equilibrium modulation behavior, and thereby, both direct kinematics and instantaneous kinematics are formulated for control purposes. To account for errors potentially caused by the simplistic modeling assumptions, a modeling uncertainty term is introduced, and the identification Jacobian along with a calibration framework to capture the parameterization is developed. Using the multi-backbone continuum robot design in [21] as a validation platform, we validate the kinematic model and model calibration experimentally while augmenting these results with additional simulation validations.

4.3 Equilibrium Modulation Backbone Insertion to Create Micro Scale Motion

This section presents the bending shape equilibrium modeling in the case where the Equilibrium Modulation Backbone (EMB) insertion is at a given depth q_s. To motivate the modeling approach taken we will first refer the readers to [86] where the simplified kinematics of multi-backbone continuum robots was presented. When the EMBs are not inserted and for proper design parameters (e.g. small spacing between the spacer disks) the continuum segment bends in a constant curvature [102]. We use this assumption to create a simplistic equilibrium model which lends itself to fast real-time computation. Since we have to account for modeling uncertainties due to friction and material parameter uncertainties, we later lump the error of the simplified model in an uncertainty term λ that will be used to produce an updated CREM model.

4.3.1 Simplistic Equilibrium Model

Figure 4.2 shows the free body diagram of a continuum segment with and without an inserted EMB. In Fig. 4.2(b), a separation plane is defined at the insertion depth q_s, dividing the segment into two subsegment - *Inserted* and *Empty*. Though not accurate, the

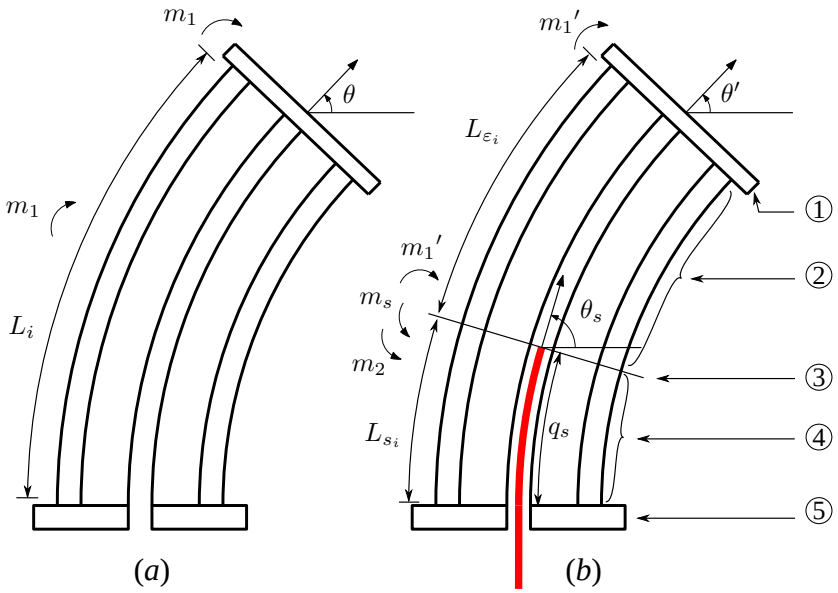


Figure 4.2: Example of a bent snake segment inserted with an equilibrium modulation backbone (EMB). For clarity, the spacer disks are not shown. ① End-disk, ② Empty subsegment, ③ Separation plane at EMB insertion depth q_s, ④ Inserted subsegment, ⑤ Base-disk.

two subsegments are both assumed to have constant but different curvatures. The angles θ' and θ_s denote the bending angles of the end-disk and at the insertion depth, respectively, when the EMB is inserted. The angle θ denotes the nominal bending angle when the EMB is not inserted. The angle $\theta_0 = \pi/2$ denotes the angle at the base of the segment.

We first consider the resultant moment m_1 that the backbones apply on any imaginary cross section of the continuum segment when no EMB is inserted (Fig. 4.2(a)):

$$m_1 = E_p I_p \frac{\theta - \theta_0}{L} + \sum_i E_i I_i \frac{\theta - \theta_0}{L_i} \tag{4.1}$$

Where E_p, E_i and I_p, I_i denote the Young's moduli and cross-sectional bending moments of inertia of the central backbone and the i^{th} secondary backbone, respectively. Also L and L_i denote the lengths of the central backbone and the i^{th} secondary backbone.

We also consider the moment $m_1{}'$ along the empty subsegment in the case of EMB

being inserted (Fig. 4.2(b)):

$$m_1{}' = E_p I_p \frac{\theta' - \theta_s}{L - q_s} + \sum_i E_i I_i \frac{\theta' - \theta_s}{L_{\varepsilon_i}} \tag{4.2}$$

Where L_{ε_i} denotes the i^{th} backbone length portion that belongs to the *empty* subsegment (this is the arc-length from the separation plane until the end-disk).

We next use key definitions from [87]. The radial distance between the secondary backbones and the primary backbone is denoted r. When r is projected onto the plane in which a segment bends, we obtain the projected radial distance Δ_i:

$$\Delta_i = r\cos(\sigma_i), \quad \sigma_i = \delta + (i-1)\beta \tag{4.3}$$

where σ_i designates the angular coordinate of the i^{th} backbone relative to the bending plane. The angular coordinate of the first backbone relative to the bending plane is δ and the angular separation between secondary backbones is $\beta = \frac{2\pi}{n}$ where n is the number of secondary backbones.

The length of the i^{th} backbone, L_i is derived using the fixed radial offset between the backbones:

$$L_i = L + \Delta_i(\theta - \theta_0) \tag{4.4}$$

Using similar rationale, we calculate the empty length portion L_{ε_i} and the inserted length portion of the i^{th} secondary backbone L_{s_i}:

$$L_{s_i} = q_s + \Delta_i(\theta_s - \theta_0) \tag{4.5}$$

$$L_{\varepsilon_i} = (L - q_s) + \Delta_i(\theta' - \theta_s) = L_i - L_{s_i} \tag{4.6}$$

In both Fig. 4.2(a) and (b), the static equilibrium at the end-disk is determined by the geometric constraints and the backbone loading forces at the end-disk. For example, coordinated pulling and pushing on all secondary backbones are assumed to form a force

couple that generates only a moment at the end-disk.

We next use a simplifying assumption that the effect of EMB wire insertion on changes in the bending curvatures of the un-inserted subsegment backbones is negligible, hence:

$$m_1 = m_1{}' \tag{4.7}$$

Next, we consider m_2 and m_s, the moments that the secondary backbones and the EMB apply on the separation plane as shown in Fig. 4.2(b):

$$m_2 = -\left(E_p I_p \frac{\theta_s - \theta_0}{q_s} + \sum_i E_i I_i \frac{\theta_s - \theta_0}{L_{s_i}}\right) \tag{4.8}$$

$$m_s = -E_s I_s \frac{\theta_s - \theta_0}{q_s} \tag{4.9}$$

Where E_s and I_s denote the Young's modulus and cross-sectional bending moment of inertia of the EMB.

Substituting equations (4.1, 4.2) into (4.7), results in one equation having two unknowns θ' and θ_s as illustrated in Fig. 4.2(b). To obtain the second equation necessary for solving for these two unknowns, we use the moment balance on the separation plane:

$$m_1{}' + m_2 + m_s = 0 \tag{4.10}$$

To solve equations (4.10) and (4.7) for the unknowns θ' and θ_s we explicitly express the backbone moments using the beam equation $m = EI\kappa$ where κ designates the radius of curvature and EI designates the bending cross sectional stiffness of a beam. In doing so, we note that the curvature of a beam bent in a circular shape satisfies $\kappa = \frac{\theta}{L}$ where θ is the deflection angle and L is the beam length. Since the backbone lengths are a function of the unknowns, we will rewrite the moment equation for a beam as $m = \frac{EI}{L}\theta$ and, by defining the beam angular deflection stiffness $k_\theta \triangleq \frac{EI}{L}$ we obtain a simple equation for the moment $m = k_\theta \theta$.

Using the above definition for beam angular deflection stiffness, we rewrite the moment equations for each beam as:

$$m_1 = k_{\theta_0}(\theta - \theta_0), \quad k_{\theta_0} = \frac{E_p I_p}{L} + \sum_i \frac{E_i I_i}{L_i} \tag{4.11}$$

$$m_1{}' = k_{\theta_1}(\theta' - \theta_s), \quad k_{\theta_1} = \frac{E_p I_p}{L - q_s} + \sum_i \frac{E_i I_i}{L_{\varepsilon_i}} \tag{4.12}$$

$$m_2 = -k_{\theta_2}(\theta_s - \theta_0), \quad k_{\theta_2} = \frac{E_p I_p}{q_s} + \sum_i \frac{E_i I_i}{L_{s_i}} \tag{4.13}$$

$$m_s = -k_{\theta_s}(\theta_s - \theta_0), \quad k_{\theta_s} = \frac{E_s I_s}{q_s} \tag{4.14}$$

Substituting Eq. (4.12)-(4.14) in Eq. (4.10) results in:

$$\theta_s = \frac{k_{\theta_1}\theta' + k_{\theta_2}\theta_0 + k_{\theta_s}\theta_0}{k_{\theta_1} + k_{\theta_2} + k_{\theta_s}} \tag{4.15}$$

Substituting equations (4.11) and (4.12) in Eq. (4.7) results in:

$$\theta' = \frac{k_0}{k_1}(\theta - \theta_0) + \theta_s = f_{\theta'}(\theta_s) \tag{4.16}$$

As a final step in the solution, we substitute the result in Eq. (4.16) in Eq. (4.15), thereby obtaining θ_s and subsequently θ'.

4.3.2 Updated CREM Model

Equations (4.15) and (4.16) present the solution to the simplistic modeling approach that is fundamentally based on Eq. (4.7) and Eq. (4.10). In addition to the simplified assumption, the current model also neglects modeling uncertainties due to frictional effects and material property uncertainties. Prior works in [36, 77] show that these uncertainties include friction and strain along the actuation lines, non-uniformly distributed load on backbones that causes shape deviations from constant-curvature bending, deviations in the cross section of the backbones during bending, and uncertainties in the properties of the

NiTi backbones.

To account for the modeling uncertainties caused by friction, material uncertainty[1], and our simplistic model, we introduce an uncertainty term λ to Eq. (4.10):

$$m_1{}' + m_2 + m_s = \lambda(q_s, \theta, \mathbf{k}_\lambda) \tag{4.17}$$

The uncertainty term λ captures effects of EMB insertion offset, bending angle uncertainty and a fixed offset:

$$\lambda(q_s, \theta, \mathbf{k}_\lambda) = k_{\lambda_0} + k_{\lambda_\theta}\,\theta + k_{\lambda_q}\,q_s; \quad \mathbf{k}_\lambda \triangleq [k_{\lambda_0}, k_{\lambda_q}, k_{\lambda_\theta}]^{\mathrm{T}} \tag{4.18}$$

The solution in Eq. (4.15) is also updated to:

$$\theta_s = \frac{k_1\theta' + k_2\theta_0 + k_s\theta_0 - \lambda}{k_1 + k_2 + k_s} \tag{4.19}$$

Having obtained the solutions to the equilibrium tip bending angle θ' and the bending angle at the separation plane θ_s, we introduce an *equilibrium configuration space* variable vector ϕ to combine them. With the purpose of preparing for kinematic derivations in later sections when we break a single continuum segment down to two subsegments, the vector ϕ is defined as:

$$\phi \triangleq [\theta_s, \theta_\varepsilon]^{\mathrm{T}}, \qquad \theta_\varepsilon \triangleq \theta' + \left(\tfrac{\pi}{2} - \theta_s\right) \tag{4.20}$$

Where θ_ε represents the bending angle of the empty subsegment.

We define the *configuration space* variable ψ as the *nominal* bending angle θ and the bending plane angle δ:

$$\psi \triangleq [\theta, \delta]^{\mathrm{T}} \tag{4.21}$$

Finally, the solution to equilibrium modeling problem is presented as a mapping $\mathcal{F}_{\mathrm{eqm}}$

[1]Manufacturer-specified Young's modulus for superelastic NiTi is typically provided with a wide range of 40-70 GPa

which is used in deriving the Jacobian matrices in the following sections:

$$\phi = \mathcal{F}_{\text{eqm}}(\psi, q_s, \mathbf{k}_\lambda), \quad \phi \in \mathbb{R}2, \psi \in \mathbb{R}2, \mathbf{k}_\lambda \in \mathbb{R}3 \tag{4.22}$$

Equation (4.22) provides the end disk equilibrium angle for a combination of any given EMB insertion length q_s, nominal bending angle θ, and bending plane angle δ.

4.4 Kinematic Modeling

Kinematic modeling of CREM includes the mapping of configuration space to task space and its differential kinematics. The differential kinematics include the instantaneous kinematics and the error propagation.

The instantaneous kinematics is derived for control purpose, and it includes two motion Jacobian matrices that both relate actuation speeds to the robot tip velocity. The *macro* motion Jacobian $\mathbf{J}_M$ is associated with the joint velocities of push/pull on backbones (direct actuation) while the *micro* motion Jacobian $\mathbf{J}_\mu$ is related to the EMB insertion velocity (indirect actuation).

The kinematic error propagation investigates how errors in parameters contribute to errors in kinematic measurements of task space (e.g. measured positions). In this work, we focus on the vector $\mathbf{k}_\lambda$ that parameterizes the modeling uncertainty. Other robot geometric kinematic parameters can be calibrated following [96]. An *identification* Jacobian $\mathbf{J}_\mathbf{k}$ is derived and used in section 4.5 to estimate $\mathbf{k}_\lambda$ with experimental measurements.

4.4.1 Kinematic Modeling Using Mapping $\mathcal{F}_{eqm}$

With the mapping $\mathcal{F}_{\text{eqm}}$ in Eq. (4.22) derived as the result of static equilibrium, the kinematic mapping can be formulated by considering a single continuum segment as two concatenated subsegments - the inserted and the empty, divided at the insertion depth q_s. Figure 4.3 illustrates our approach to analyzing the two concatenated subsegments. The bending angles of both subsegments were introduced in Eq. (4.20), denoted as θ_s and θ_ε,

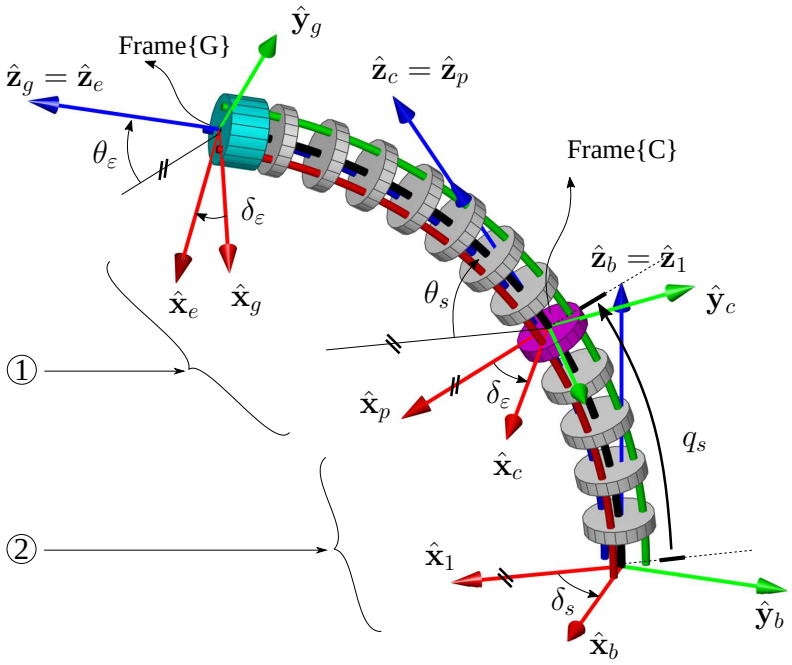


Figure 4.3: A single segment treated as two concatenated subsegments for a given micro-motion wire insertion depth. ① Empty subsegment, ② Inserted subsegment.

for the inserted and the empty subsegment. Since the whole segment is assumed to bend in plane, both subsegments have the equal bending plane angles:

$$\delta_s = \delta_\varepsilon = \delta \tag{4.23}$$

The kinematic nomenclature used in this paper refers to Table 4.1 (shown in Fig. 4.3).

Recalling the direct kinematics of a single segment [36] having length L_x and an end disk angle θ_x, the end disk pose (position and orientation) relative to the base are given by:

$${}^{\text{base}}\mathbf{p}_{\text{end-disk / base}} = \frac{L_x}{\theta_x - \pi/2} \begin{bmatrix} \cos\delta_x\,(\sin\theta_x - 1) \\ -\sin\delta_x\,(\sin\theta_x - 1) \\ -\cos\theta_x \end{bmatrix} \tag{4.24}$$

$${}^{\text{base}}\mathbf{R}_{\text{end-disk}} = e^{-\delta_x[\mathbf{z}^\wedge]}\; e^{(\frac{\pi}{2}-\theta_x)[\mathbf{y}^\wedge]}\; e^{\delta_x[\mathbf{z}^\wedge]} \tag{4.25}$$

Where δ_x designates the angle of the bending plane (analogous to δ in Fig. 4.3), $[\mathbf{v}^\wedge]$ represents the cross-product matrix of vector $\mathbf{v}$ and the matrix exponential $e^{\alpha[\mathbf{v}^\wedge]}$ represents a rotation matrix about the axis $\mathbf{v}$ by an angle α.

To obtain the pose of the end disk of the inserted segment is given by ${}^b\mathbf{p}_{c/b}$ and ${}^b\mathbf{R}_c$ we substitute $L_x = q_s, \theta_x = \theta_s, \delta_x = \delta$ in Eqs. (4.24, 4.25). Similarly, the pose of the end disk of the empty segment relative to its base is obtained by substituting $L_x = L - q_s, \theta_x = \theta_\varepsilon, \delta_x = \delta$ in Eqs. (4.24, 4.25) to result in ${}^c\mathbf{p}_{g/c}$ and ${}^c\mathbf{R}_g$.

The pose of the free subsegment end disk relative to the segment base is given by:

$$ {}^b\mathbf{p}_{g/b} = {}^b\mathbf{p}_{c/b} + {}^b\mathbf{R}_c\, {}^c\mathbf{p}_{g/c} \triangleq \mathcal{F}_g(\phi, \delta, q_s) \tag{4.26} $$

$$ {}^b\mathbf{R}_g = {}^b\mathbf{R}_c\, {}^c\mathbf{R}_b = \mathbf{R}_g(\phi, \delta) \tag{4.27} $$

Casting the above two equations in a homogeneous transform format yields:

$$ {}^b\mathbf{T}_g = \begin{bmatrix} {}^b\mathbf{R}_g & {}^b\mathbf{p}_{g/b} \\ \mathbf{0} & 1 \end{bmatrix} \triangleq \mathcal{F}_T(\phi, \delta, q_s) \tag{4.28} $$

With ϕ expressed using mapping $\mathcal{F}_{\text{eqm}}(\psi, q_s, \mathbf{k}_\lambda)$, the forward kinematics is determined, which can be also written as:

$$ {}^b\mathbf{T}_g \triangleq \mathcal{F}_T(\psi, q_s, \mathbf{k}_\lambda) \tag{4.29} $$

4.4.2 Differential Kinematics

The total differential of a homogenous transformation $\mathbf{T} \in \mathrm{SE}(3)$, may be represented as:

$$ \mathrm{d}\boldsymbol{\xi} \triangleq \left[(\mathrm{d}\mathbf{x})^{\mathrm{T}}, (\mathrm{d}\boldsymbol{\mu})^{\mathrm{T}}\right]^{\mathrm{T}}, \quad \mathrm{d}\boldsymbol{\xi} \in \mathbb{R}6 \times 1 \tag{4.30} $$

$$ \mathrm{d}\mathbf{x} \triangleq \mathrm{d}({}^b\mathbf{p}_{g/b}), \quad \mathrm{d}\boldsymbol{\mu} \triangleq [\mathrm{d}\mu_x, \mathrm{d}\mu_y, \mathrm{d}\mu_z]^{\mathrm{T}} \tag{4.31} $$

Where d$\mathbf{x}$ and d$\boldsymbol{\mu}$ represent translational and rotational differentials in the base frame[2]. The vector $\boldsymbol{\mu} \in \mathbb{R}3 \times 1$ represents a chosen orientation parametrization (e.g. Euler angles).

The total differential of ${}^{b}\mathbf{T}_{g}$ is obtained by considering differentials on all variables, i.e. dϕ, dδ, and dq_s:

$$\mathrm{d}\boldsymbol{\xi} = \frac{\partial \boldsymbol{\xi}}{\partial \phi}\,\mathrm{d}\phi + \frac{\partial \boldsymbol{\xi}}{\partial \delta}\,\mathrm{d}\delta + \frac{\partial \boldsymbol{\xi}}{\partial q_s}\,\mathrm{d}q_s \tag{4.32}$$

Using the nomenclature of a Jacobian $\mathbf{J}_{ab}$ such that $\delta a = \mathbf{J}_{ab}\delta b$, we define the following Jacobian matrices:

$$\frac{\partial \boldsymbol{\xi}}{\partial \phi} \triangleq \mathbf{J}_{\boldsymbol{\xi}\phi} \in \mathbb{R}6 \times 2, \quad \frac{\partial \boldsymbol{\xi}}{\partial \delta} \triangleq \mathbf{J}_{\boldsymbol{\xi}\delta} \in \mathbb{R}6, \quad \frac{\partial \boldsymbol{\xi}}{\partial q_s} \triangleq \mathbf{J}_{\boldsymbol{\xi}q_s} \in \mathbb{R}6 \tag{4.33}$$

The Jacobian matrices $\mathbf{J}_{\boldsymbol{\xi}\phi}$, $\mathbf{J}_{\boldsymbol{\xi}\delta}$, and $\mathbf{J}_{\boldsymbol{\xi}q_s}$, respectively, relate the differential on equilibrium configuration space variable $\mathrm{d}\phi = [\mathrm{d}\theta_s, \mathrm{d}\theta_\varepsilon]^{\mathrm{T}}$, the differential on bending plane angle dδ, and differential on EMB insertion depth dq_s, to the corresponding differential contributions on the pose vector d$\boldsymbol{\xi}$.

Both $\mathbf{J}_{\boldsymbol{\xi}\phi}$ and $\mathbf{J}_{\boldsymbol{\xi}\delta}$ can be obtained by treating the inserted and empty subsegments as a concatenated two-segment robot, which is explained in section 4.4.3.

The third Jacobian, $\mathbf{J}_{\boldsymbol{\xi}q_s}$, defined as the partial derivative, $\partial\boldsymbol{\xi}/\partial q_s$, is derived with the other variables (ϕ and δ) held constant. The end-effector orientation, given by ${}^{b}\mathbf{R}_{g}$ in Eq. (4.27) is not a function of q_s. Therefore, by considering only the translational differential due to dq_s, we have:

$$\mathbf{J}_{\boldsymbol{\xi}q_s} = \begin{bmatrix} \dfrac{\partial\, {}^{b}\mathbf{p}_{c/b}}{\partial q_s} + {}^{b}\mathbf{R}_{c}\,\dfrac{\partial\, {}^{c}\mathbf{p}_{g/c}}{\partial q_s} \\ \hdashline \mathbf{0}_{3\times 1} \end{bmatrix} \tag{4.34}$$

Where $\frac{\partial\, {}^{b}\mathbf{p}_{c/b}}{\partial q_s}$ and $\frac{\partial\, {}^{c}\mathbf{p}_{g/c}}{\partial q_s}$ are derived from Eq. (4.24). It is important to note that $\mathbf{J}_{\boldsymbol{\xi}q_s}$ differs from the micro motion Jacobian $\mathbf{J}_{\mu}$ derived later in that $\mathbf{J}_{\boldsymbol{\xi}q_s}$ is a contributing part of

[2] A differential rotation is a sequence of rotations of small angles.

Table 4.1: Nomenclature for Kinematic Modeling

Symbol	Description
Frame $\{\text{F}\}$	designates a right-handed frame with unit vectors $\hat{\mathbf{x}}_f, \hat{\mathbf{y}}_f, \hat{\mathbf{z}}_f$ and point $\mathbf{f}$ as its origin.
Frame $\{\text{B}\}$	the base disk frame with $\mathbf{b}$ located at the center of the base disk, $\hat{\mathbf{x}}_b$ passing through the first secondary backbone and $\hat{\mathbf{z}}_b$ perpendicular to the base disk.
Frame $\{1\}$	Frame of the bending plane having $\hat{\mathbf{z}}_1 = \hat{\mathbf{z}}_b$ and $\hat{\mathbf{x}}_1$ passing through with the project point of the end disk center. The angle δ is defined as from $\hat{\mathbf{x}}_1$ to $\hat{\mathbf{x}}_b$ about $\hat{\mathbf{z}}_b$ according to right hand rule.
Frames $\{\text{E}\}$ & $\{\text{G}\}$	Frame $\{\text{E}\}$ is defined with $\hat{\mathbf{z}}_e$ as the normal to the *end* disk and $\hat{\mathbf{x}}_e$ is the intersection of the bending plane and the end disk top surface. Frame $\{\text{G}\}$ is the *gripper* frame that has the same $\hat{\mathbf{z}}$ as $\{\text{E}\}$, i.e. $\hat{\mathbf{z}}_g = \hat{\mathbf{z}}_e$, but $\hat{\mathbf{x}}_g$ is passing through the first secondary backbone. It can be obtained by a rotation angle $(-\sigma_{1e})$ about $\hat{\mathbf{z}}_e$.
Frames $\{\text{P}\}$ & $\{\text{C}\}$	These frames are defined in a manner similar to the definition of frames $\{\text{E}\}$ and $\{\text{G}\}$ but for a specific arc insertion length q_s as opposed to the full length of the robot segment L. The $\hat{\mathbf{x}}_c - \hat{\mathbf{y}}_c$ plane is the insertion plane as in shown in the planar case in Fig. 4.2.
Frame $\{\text{I}\}$	designates the microscope image frame having the origin at the corner of the image and having its x-y axes aligned with the width and height directions $\big($Fig. 4.8(a, c)$\big)$.
Frame $\{\text{M}\}$	designates the marker frame that is determined by segmentation of the microscope image $\big($Fig. 4.8(c)$\big)$.
Vector ${}^{x}\mathbf{p}_{a/b}$	designates the position of point a relative to point b that is expressed in frame $\{\text{X}\}$.

$\mathbf{J}_\mu$ - the length 'tangential' contribution, while $\mathrm{d}q_s$ also propagates to $\mathrm{d}\phi$ that also causes change on $\mathrm{d}\boldsymbol{\xi}$.

Having derived the above three Jacobian matrices, $\mathbf{J}_{\boldsymbol{\xi}\phi}$, $\mathbf{J}_{\boldsymbol{\xi}\delta}$, and $\mathbf{J}_{\boldsymbol{\xi}q_s}$, we obtain the pose total differential $\mathrm{d}\boldsymbol{\xi}$ expressed using differentials $\mathrm{d}\phi$, $\mathrm{d}\delta$, and $\mathrm{d}q_s$. Further, the differential $\mathrm{d}\phi$ is a result of multiple other differentials, which can be seen from mapping $\mathcal{F}_{\text{eqm}}(\psi, q_s, \mathbf{k}_\lambda)$. To fully investigate and decouple the contributions of direct (macro) and indirect (micro) actuation, we express $\mathrm{d}\phi$ using differentials on $(\psi, q_s, \mathbf{k}_\lambda)$. This differentiation is also motivated by Eq. (4.29), where the variables are decoupled as ψ for macro motion, q_s for

micro motion, and $\mathbf{k}_\lambda$ for micro motion parameters. Such differentiation is derived as:

$$\mathrm{d}\phi = \frac{\partial \phi}{\partial \psi}\mathrm{d}\psi + \frac{\partial \phi}{\partial q_s}\mathrm{d}q_s + \frac{\partial \phi}{\partial \mathbf{k}_\lambda}\mathrm{d}\mathbf{k}_\lambda \tag{4.35}$$

$$\frac{\partial \phi}{\partial \psi} = \left[\frac{\partial \phi}{\partial \theta}, \frac{\partial \phi}{\partial \delta}\right], \quad \frac{\partial \phi}{\partial \mathbf{k}_\lambda} = \left[\frac{\partial \phi}{\partial k_{\lambda_0}}, \frac{\partial \phi}{\partial k_{\lambda_\theta}}, \frac{\partial \phi}{\partial k_{\lambda_q}}\right] \tag{4.36}$$

Where the gradient terms are derived in Appendix E.1 as:

$$\frac{\partial \phi}{\partial \theta} = (\mathbf{A}\mathbf{S}_0 - \mathbf{\Gamma}_{\theta_s}\mathbf{S}_1)^{-1}\ \mathbf{\Gamma}_\theta \triangleq \mathbf{J}_{\phi\theta} \tag{4.37}$$

$$\frac{\partial \phi}{\partial \delta} = (\mathbf{A}\mathbf{S}_0 - \mathbf{\Gamma}_{\theta_s}\mathbf{S}_1)^{-1}\ \mathbf{\Gamma}_\delta \triangleq \mathbf{J}_{\phi\delta} \tag{4.38}$$

$$\frac{\partial \phi}{\partial q_s} = (\mathbf{A}\mathbf{S}_0 - \mathbf{\Gamma}_{\theta_s}\mathbf{S}_1)^{-1}\ \mathbf{\Gamma}_{q_s} \triangleq \mathbf{J}_{\phi q_s} \tag{4.39}$$

$$\frac{\partial \phi}{\partial k_{\lambda_i}} = (\mathbf{A}\mathbf{S}_0 - \mathbf{\Gamma}_{\theta_s}\mathbf{S}_1)^{-1}\ \mathbf{B}'_{k_{\lambda_i}} \triangleq \mathbf{J}_{\phi k_{\lambda_i}},\ \frac{\partial \phi}{\partial \mathbf{k}_\lambda} \triangleq \mathbf{J}_{\phi \mathbf{k}_\lambda} \tag{4.40}$$

$$\mathbf{C}_\phi \triangleq \mathbf{S}_0\phi - \mathbf{C}_0, \quad \mathbf{\Gamma}_x = \mathbf{B}'_x - \mathbf{A}'_x\mathbf{C}_\phi \tag{4.41}$$

Matrices $\mathbf{A}'_x, \mathbf{B}'_x$ are partial derivative matrices with respect to variable 'x', and $\mathbf{A}, \mathbf{S}_0, \mathbf{B}, \mathbf{C}_0, \mathbf{S}_1$ are defined as:

$$\mathbf{A} = \begin{bmatrix} k_1 + k_2 + k_s & -k_1 \\ k_1 & -k_1 \end{bmatrix},\ \mathbf{S}_0 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \tag{4.42}$$

$$\mathbf{B} = \begin{bmatrix} (k_2 + k_s)\,\theta_0 - \lambda \\ k_0(\theta_0 - \theta) \end{bmatrix},\ \mathbf{C}_0 = \begin{bmatrix} 0 \\ \theta_0 \end{bmatrix},\ \mathbf{S}_1 = \begin{bmatrix} 1 & 0 \end{bmatrix} \tag{4.43}$$

Using Eq. (4.37-4.43), $\mathrm{d}\phi$ is fully expressed as Eq. (4.35). Substituting $\mathrm{d}\phi$ into the original differentiation in Eq. (4.32), we obtain the full differential kinematics that relates differentials on $\{\psi, q_s, \mathbf{k}_\lambda\}$ to the pose total differential $\mathrm{d}\boldsymbol{\xi}$:

$$\begin{aligned} \mathrm{d}\boldsymbol{\xi} = \ & \frac{\partial \boldsymbol{\xi}}{\partial \phi}\frac{\partial \phi}{\partial \theta}\mathrm{d}\theta + \frac{\partial \boldsymbol{\xi}}{\partial \phi}\frac{\partial \phi}{\partial \delta}\mathrm{d}\delta + \frac{\partial \boldsymbol{\xi}}{\partial \phi}\frac{\partial \phi}{\partial q_s}\mathrm{d}q_s + \\ & \frac{\partial \boldsymbol{\xi}}{\partial \phi}\frac{\partial \phi}{\partial \mathbf{k}_\lambda}\mathrm{d}\mathbf{k}_\lambda + \frac{\partial \boldsymbol{\xi}}{\partial \delta}\mathrm{d}\delta + \frac{\partial \boldsymbol{\xi}}{\partial q_s}\mathrm{d}q_s \end{aligned} \tag{4.44}$$

Rewriting Eq. (4.45) using the Jacobian definitions:

$$\begin{aligned} \mathrm{d}\boldsymbol{\xi} = \; & \mathbf{J}_{\xi\phi}\mathbf{J}_{\phi\theta}\,\mathrm{d}\theta \;+\; \mathbf{J}_{\xi\phi}\mathbf{J}_{\phi\delta}\,\mathrm{d}\delta \;+\; \mathbf{J}_{\xi\delta}\,\mathrm{d}\delta \;+ \\ & \mathbf{J}_{\xi\phi}\mathbf{J}_{\phi q_s}\,\mathrm{d}q_s \;+\mathbf{J}_{\xi q_s}\,\mathrm{d}q_s \;+\mathbf{J}_{\xi\phi}\mathbf{J}_{\phi\mathbf{k}_\lambda}\,\mathrm{d}\mathbf{k}_\lambda \end{aligned} \tag{4.45}$$

Collecting like terms of $\mathrm{d}\psi$, $\mathrm{d}q_s$, and $\mathrm{d}\mathbf{k}_\lambda$, we obtain a differentiation that decouples differentials of the macro motion, the micro motion, and the parameters:

$$\begin{aligned} \mathrm{d}\boldsymbol{\xi} = \; & \underbrace{\left[\begin{array}{c:c} \mathbf{J}_{\xi\phi}\mathbf{J}_{\phi\theta} & \mathbf{J}_{\xi\phi}\mathbf{J}_{\phi\delta}+\mathbf{J}_{\xi\delta} \end{array}\right]}_{\triangleq\, \mathbf{J}_{M\psi}}\mathrm{d}\psi \;+ \\ & \underbrace{\left[\mathbf{J}_{\xi\phi}\mathbf{J}_{\phi q_s}+\mathbf{J}_{\xi q_s}\right]}_{\triangleq\, \mathbf{J}_{\mu}}\mathrm{d}q_s \;+\; \underbrace{\mathbf{J}_{\xi\phi}\mathbf{J}_{\phi\mathbf{k}_\lambda}}_{\triangleq\, \mathbf{J}_{\mathbf{k}}}\mathrm{d}\mathbf{k}_\lambda \end{aligned} \tag{4.46}$$

The above result completes the mapping from configuration to task space. It clearly delineates the effects of EDM insertion and direct actuation to achieving macro and micro motion. For control purposes, a complete mapping from joint to task space is needed. We therefore consider next the mapping from direct (macro) actuation joint space $\mathbf{q}$ to task space $\boldsymbol{\xi}$. Since three secondary backbones are used in our experimental setup as illustrated in Figure 4.3, we will define $\mathbf{q} \triangleq [q_1, q_2, q_3]^{\mathrm{T}}$ where:

$$q_i \triangleq L_i - L \tag{4.47}$$

When obtaining this mapping, we consider the nominal segment kinematics for multi-backbone continuum robots as in [86].

The Jacobian that relates the differential $\mathrm{d}\mathbf{q}$ to the differential $\mathrm{d}\psi$ was reported in [102] as:

$$\mathrm{d}\mathbf{q} \triangleq \mathbf{J}_{\mathbf{q}\psi}\,\mathrm{d}\psi, \quad \mathbf{J}_{\mathbf{q}\psi} = r\begin{bmatrix} c_\delta & (\theta_0-\theta)\,s_\delta \\ c_{(\delta+\beta)} & (\theta_0-\theta)\,s_{(\delta+\beta)} \\ c_{(\delta+2\beta)} & (\theta_0-\theta)\,s_{(\delta+2\beta)} \end{bmatrix} \tag{4.48}$$

Where r denotes the constant radial distance between the central and surrounding backbones, and $\beta = 2\pi/3$ denotes the backbone separation angle. Using Eq. (4.48), we substitute $\mathrm{d}\psi$ into Eq. (4.46), arriving at the final differential kinematics:

$$\mathrm{d}\boldsymbol{\xi} = \mathbf{J}_M \, \mathrm{d}\mathbf{q} + \mathbf{J}_\mu \, \mathrm{d}q_s + \mathbf{J}_\mathbf{k} \, \mathrm{d}\mathbf{k}_\lambda \tag{4.49}$$

Equation (4.49) fully decouples the end-effector pose differential to contributions of the direct (macro) actuation $\mathrm{d}\mathbf{q}$, the indirect (micro) actuation $\mathrm{d}q_s$, and the modeling uncertainty $\mathrm{d}\mathbf{k}_\lambda$. The three Jacobian matrices are obtained from Eq. (4.49) as an important finding of this paper: $\mathbf{J}_M$ defined as the *Macro* motion Jacobian, $\mathbf{J}_\mu$ defined as the *Micro* motion Jacobian, and $\mathbf{J}_\mathbf{k}$ defined as the *Identification* Jacobian.

$$\mathbf{J}_M = \left[\begin{array}{c:c} \mathbf{J}_{\xi\phi}\mathbf{J}_{\phi\theta} & \mathbf{J}_{\xi\phi}\mathbf{J}_{\phi\delta} + \mathbf{J}_{\xi\delta} \end{array} \right] (\mathbf{J}_{\mathbf{q}\psi})^\dagger, \; \mathbf{J}_M \in \mathbb{R}6 \times 2 \tag{4.50}$$

$$\mathbf{J}_\mu = \mathbf{J}_{\xi\phi}\mathbf{J}_{\phi q_s} + \mathbf{J}_{\xi q_s}, \qquad \mathbf{J}_\mu \in \mathbb{R}6 \times 1 \tag{4.51}$$

$$\mathbf{J}_\mathbf{k} = \mathbf{J}_{\xi\phi}\mathbf{J}_{\phi\mathbf{k}_\lambda}, \qquad \mathbf{J}_\mathbf{k} \in \mathbb{R}6 \times n_k \tag{4.52}$$

Where $(\cdot)^\dagger$ is the Moore-Penrose pseudo inverse.

4.4.3 Deriving $\mathbf{J}_{\xi\phi}$ and $\mathbf{J}_{\xi\delta}$

The result in Eq. (4.49) builds on knowing the Jacobians $\mathbf{J}_{\xi\phi} \in \mathbb{R}6 \times 2$ and $\mathbf{J}_{\xi\delta} \in \mathbb{R}6 \times 1$ as mentioned in Eqs. (4.32, 4.33). We now provide a derivation to these two Jacobians. Considering a single-segment CREM as two concatenated subsegments (inserted and empty), we apply the Jacobian formulation for a two-segment multi-backbone continuum robot (MBCR) while assuming that both subsegments share the bending plane angle δ. For the ease of adapting formulations from [102], we introduce a vector notation:

$$\boldsymbol{\delta}_v \triangleq \begin{bmatrix} \delta_s \\ \delta_\varepsilon \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \delta, \quad \mathrm{d}\boldsymbol{\delta}_v \triangleq \begin{bmatrix} \mathrm{d}\delta_s \\ \mathrm{d}\delta_\varepsilon \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \mathrm{d}\delta \tag{4.53}$$

We next use the notation of ${}^{i-1}\boldsymbol{\xi}_{i/i-1}$ to denote the pose of the i^{th} subsegment relative to the $(i-1)^{\text{th}}$ subsegment where $i \in \{s, \varepsilon\}$. Using $\mathbf{v}$ and $\boldsymbol{\omega}$ to denote linear and angular velocities, we define the corresponding four Jacobian matrices corresponding with the contributions of $\mathrm{d}\theta_i, \mathrm{d}\delta_i$ where $i \in \{s, \varepsilon\}$ to the end-effector twist:

$$\frac{\partial\, {}^{i-1}\boldsymbol{\xi}_{i/i-1}}{\partial\left([\theta_i, \delta_i]^{\mathrm{T}}\right)} \triangleq \left[\begin{array}{c:c} \mathbf{J}_{\mathbf{v}\theta_i} & \mathbf{J}_{\mathbf{v}\delta_i} \\ \hdashline \mathbf{J}_{\boldsymbol{\omega}\theta_i} & \mathbf{J}_{\boldsymbol{\omega}\delta_i} \end{array}\right] \in \mathbb{R}6 \times 2, \quad i \in \{s, \varepsilon\} \tag{4.54}$$

Details of the derivations of the Jacobians are provided in Appendix E.2.

Following [103], the serial composition of two subsegments using twist transformation results in the end effector twist:

$$\mathbf{J}_{\xi\phi} = \frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\phi}} = \left[\begin{array}{c:c} \mathbf{J}_{\mathbf{v}\theta_s} - \left[{}^{b}\mathbf{R}_c\, {}^{c}\mathbf{p}_{g/c}\right]^{\wedge} \mathbf{J}_{\boldsymbol{\omega}\theta_s} & {}^{b}\mathbf{R}_c \mathbf{J}_{\mathbf{v}\theta_\varepsilon} \\ \hdashline \mathbf{J}_{\boldsymbol{\omega}\theta_s} & {}^{b}\mathbf{R}_c \mathbf{J}_{\boldsymbol{\omega}\theta_\varepsilon} \end{array}\right] \tag{4.55}$$

$$\mathbf{J}_{\xi\delta_v} = \frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\delta}_v} = \left[\begin{array}{c:c} \mathbf{J}_{\mathbf{v}\delta_s} - \left[{}^{b}\mathbf{R}_c\, {}^{c}\mathbf{p}_{g/c}\right]^{\wedge} \mathbf{J}_{\boldsymbol{\omega}\delta_s} & {}^{b}\mathbf{R}_c \mathbf{J}_{\mathbf{v}\delta_\varepsilon} \\ \hdashline \mathbf{J}_{\boldsymbol{\omega}\delta_s} & {}^{b}\mathbf{R}_c \mathbf{J}_{\boldsymbol{\omega}\delta_\varepsilon} \end{array}\right] \tag{4.56}$$

These definitions of $\mathbf{J}_{\xi\phi}$ and $\mathbf{J}_{\xi\delta_v}$ complete the two missing terms needed in Eq. (4.49), but with a slight formulation modification. The Jacobian matrix $\mathbf{J}_{\xi\delta}$ is slightly different from $\mathbf{J}_{\xi\delta_v}$ in Eq. (4.56), and using the differentiation chain rule it becomes:

$$\mathbf{J}_{\xi\delta} \triangleq \frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\delta}} = \frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\delta}_v} \frac{\mathrm{d}\boldsymbol{\delta}_v}{\mathrm{d}\boldsymbol{\delta}} = \mathbf{J}_{\xi\delta_v} \begin{bmatrix} 1 \\ & 1 \end{bmatrix} \tag{4.57}$$

4.5 Calibration of Micro Motion Parameters

To calibrate the model uncertainty parameters $\mathbf{k}_\lambda$, we extract from Eq. (4.49) the following relation:

$$\delta\boldsymbol{\xi}(\delta\mathbf{k}_\lambda) = \mathbf{J}_{\mathbf{k}} \delta\mathbf{k}_\lambda \tag{4.58}$$

Using this error propagation model, we construct an estimation method to estimate $\mathbf{k}_\lambda$. Let $\boldsymbol{\xi}_j \leftrightarrow \left[\bar{\mathbf{x}}_j, \bar{\mathbf{R}}_j\right]$ designate the measured end-effector pose at the j^{th} robot configuration (insertion depth) where $\bar{\mathbf{x}}_j$ and $\bar{\mathbf{R}}_j$ designate the measured position and orientation. Let $\mathbf{x}_j$ and $\mathbf{R}_j$ denote the modeled pose using the direct kinematics as presented in section 4.4.1 for a given $\mathbf{k}_\lambda$. The error between the measured and modeled poses are then defined as:

$$\mathbf{c}_j \triangleq \left[(\bar{\mathbf{x}}_j - \mathbf{x}_j)^{\mathrm{T}}, (\alpha_{e_j}\hat{\mathbf{m}}_{e_j})^{\mathrm{T}}\right]^{\mathrm{T}}, \quad \mathbf{c}_j \in \mathbb{R}6 \tag{4.59}$$

where α_{e_j} and $\hat{\mathbf{m}}_{e_j}$ are the angle and axis parameterizing the orientation error $\mathbf{R}_{e_j}$. These parameters are given by:

$$\mathbf{R}_{e_j} \triangleq \bar{\mathbf{R}}_j {\mathbf{R}_j}^{\mathrm{T}} = e^{\alpha_{e_j}[\hat{\mathbf{m}}_{e_j}]^\wedge} \tag{4.60}$$

$$\alpha_{e_j} = \cos^{-1}\left(\frac{\mathrm{Tr}(\mathbf{R}_{e_j})-1}{2}\right) \tag{4.61}$$

$$\hat{\mathbf{m}}_{e_j} = \frac{1}{2\sin(\alpha_e)}\left(\mathbf{R}_{e_j} - {\mathbf{R}_{e_j}}^{\mathrm{T}}\right)^\vee \tag{4.62}$$

where the operator $(\cdot)^\vee$ designates the vector form of a skew-symmetric matrix.

An *aggregated* error vector $\widetilde{\mathbf{c}}_\lambda$ is defined to include errors of all N robot configurations:

$$\widetilde{\mathbf{c}}_\lambda = \left[(\mathbf{c}_1)^{\mathrm{T}}, \ldots, (\mathbf{c}_N)^{\mathrm{T}}\right]^{\mathrm{T}} \tag{4.63}$$

The optimization objective function M_λ is then defined as:

$$M_\lambda(\mathbf{k}_\lambda) = \frac{1}{2N}\,{\widetilde{\mathbf{c}}_\lambda}^{\mathrm{T}}\,\mathbf{W}\,\widetilde{\mathbf{c}}_\lambda \tag{4.64}$$

Where $\mathbf{W}$ is a weight matrix encoding confidence in the measurements and the measurement unit scaling factors.

The first-order Taylor series approximation of M_λ is given:

$$M_\lambda(\mathbf{k}_\kappa + \delta\mathbf{k}_\kappa) \approx M_\lambda(\mathbf{k}_\lambda) + \mathbf{J}_{M_\lambda}\delta\mathbf{k}_\lambda \tag{4.65}$$

where *the aggregated* Jacobian $\mathbf{J}_{M_\lambda} \in \mathbb{R}1 \times 5$ is given by:

$$\mathbf{J}_{M_\lambda} = \frac{1}{N}(\widetilde{\mathbf{c}}_\lambda)^{\mathrm{T}}\,\mathbf{W}\,\mathbf{J}_{c_\lambda} \tag{4.66}$$

$$\mathbf{J}_{c_\lambda} = \frac{\partial\widetilde{\mathbf{c}}_\lambda}{\partial\mathbf{k}_\lambda} = -\left[(\mathbf{J}_{\mathbf{k}_1})^{\mathrm{T}}, \ldots, (\mathbf{J}_{\mathbf{k}_N})^{\mathrm{T}}\right]^{\mathrm{T}} \tag{4.67}$$

Equation (4.66) shows that minimizing M_λ entails following the gradient descent direction along $(\partial\widetilde{\mathbf{c}}_\lambda/\partial\mathbf{k}_\lambda)$. The parameter $\mathbf{k}_\lambda$ is then obtained using an iterative nonlinear least squares estimation shown in Algorithm 4.

Algorithm 4 Nonlinear LS Estimate

Require: $\mathscr{D}\{(\bar{\mathbf{x}}_j, \psi_j, q_{s_j})\}_{,\ j=1,\ldots,N}$; $\mathbf{k}_{\lambda_0}$, $(\beta, \eta) > 0$

1: **START** Initialize: $\mathbf{k}_i \leftarrow \mathbf{k}_{\lambda_0}$, $M_{i-1} \leftarrow 1$, $M_i \leftarrow 100$

2: **while** $\frac{\|M_{\lambda i} - M_{\lambda i-1}\|}{M_{\lambda i-1}} \geq \beta$ **do**

3: $\quad M_{\lambda i-1} \leftarrow M_{\lambda i}$, $\widetilde{\mathbf{c}}_p = \widetilde{\mathbf{c}}_\lambda(\mathbf{k}_i)$, $M_{\lambda i} = M_{\lambda i}(\mathbf{k}_i)$,

4: $\quad \mathbf{J}_{c_\lambda} = \mathbf{J}_{c_\lambda}(\mathbf{k}_i)$

5: $\quad$ Update $\mathbf{k}_{i+1}$:

$$\mathbf{k}_{i+1} = \mathbf{k}_i - \mathbf{H}\left(\eta\,(\mathbf{J}_{c_\lambda})^{+}\,\widetilde{\mathbf{c}}_\lambda\right), \eta \in (0,1] \tag{4.68}$$

$$(\mathbf{J}_{c_\lambda})^{+} = \left((\mathbf{J}_{c_\lambda})^{\mathrm{T}}\mathbf{W}\mathbf{J}_{c_\lambda}\right)^{-1}(\mathbf{J}_{c_\lambda})^{\mathrm{T}}\mathbf{W} \tag{4.69}$$

6: **end while**

7: $\mathbf{k}^* \leftarrow \mathbf{k}_i$

Ensure: $\mathbf{k}^*$

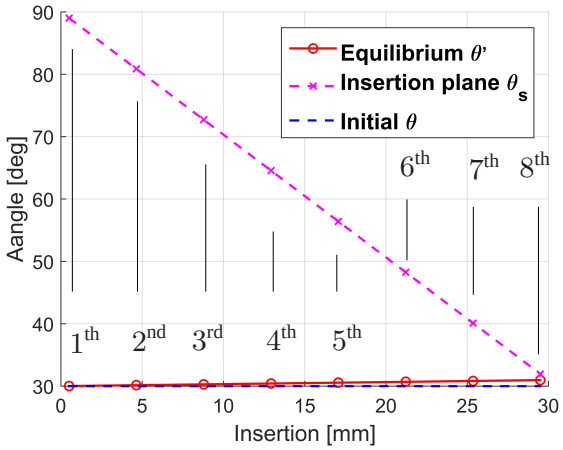
In the algorithm, $\mathbf{H}$ is the parameter scaling matrix and the task space variable scaling is achieved by adjusting $\mathbf{W}$, both of witch are discussed in details in [85].

4.6 Simulation Study of Direct Kinematics and Differential Kinematics

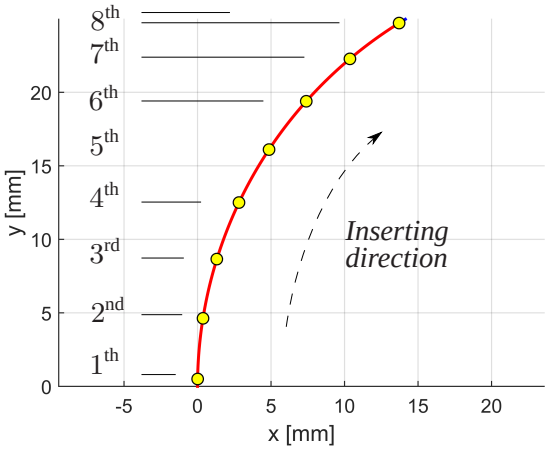
In this section, we present simulations to demonstrate the direct kinematics and differential kinematics. We also verify the differential kinematics through finite-difference simulations.

We also carry out simulations to verify the differential kinematics model. In all simulations, we assumed the robot points vertically down at its home (straight) configuration.

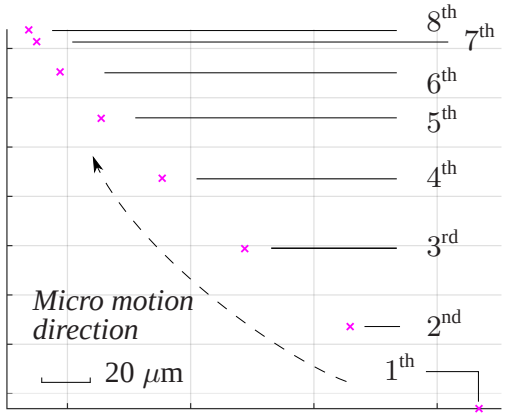
4.6.1 Position Analysis of Micro Motion



(a) calculations of the equilibrium configuration space variables $\{\theta', \theta_s\}$

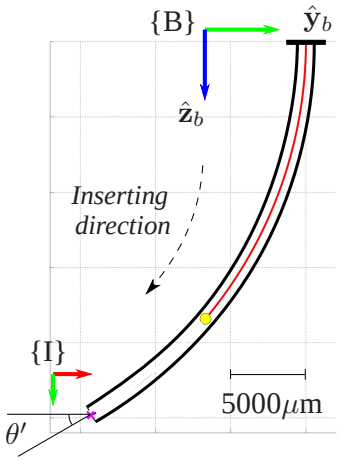


(b) the robot bending with EMB insertion sample depths

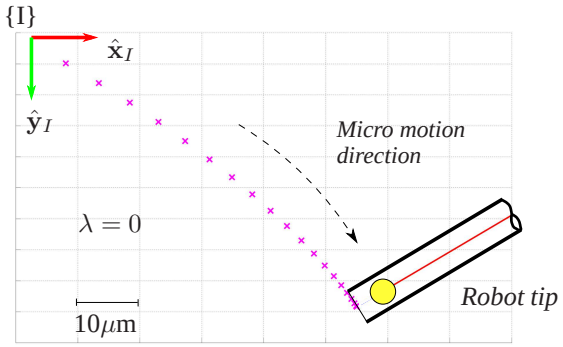


(c) zoom-in view of the tip positions that corresponds to the EMB insertion sample depths

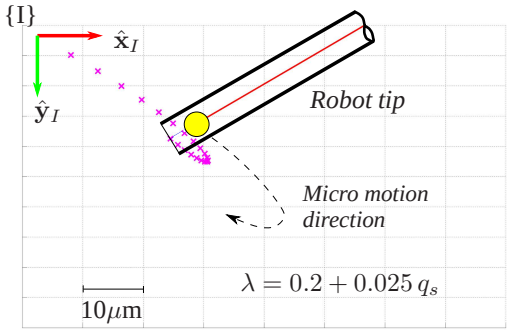
Figure 4.4: Simulations of robot equilibrium modulation backbone (EMB) insertion - (b) and (c) together show the correspondence between the EMB insertion locations and the tip positions in a zoom-view



(a) the entire segment when EMB being inserted (only central backbone and EMB are shown)



(b) zoom-in view of the tip micro motion assuming ideal condition, i.e. $\lambda = 0$



(c) zoom-in view of the tip micro motion assuming a linear uncertainty function of λ. {B} and {I} designate the robot base and the camera image frames

Figure 4.5: Simulations of continuum robot micro motion created by EMB insertion.

Using the model in section 4.4.1, we present the simulated position analysis of the micro motion created by the EMB insertion. In both simulations and the experimental model validations, we use the parameters as in Table 4.2. They include the Young's modulus of the superelastic NiTi material used for backbone tubes and EMBs (E_p, E_i, E_s), the diameters of backbones (d_p, d_i, d_s), and the cross-sectional moment of inertia (I_p, I_i, I_s).

Figure 4.5 shows the simulation results of the micro motion created by EMB insertion. Figure 4.5(a) shows the continuum segment at its initial bending angle $\theta = 30^\circ$. During

Table 4.2: Robot Parameters Used in Simulations and Experiments

L	r	E_p, E_i, E_s	d_p, d_i	d_s	I_p, I_i	I_s
44.3 mm	3 mm	41 GPa	0.90 mm	0.38 mm	0.0312 mm^4	0.0010 mm^4

simulations, the equilibrium bending angles $\{\theta', \theta_s\}$ were computed at different EMB insertion depths, shown as Fig. 4.4(a). Figure 4.4 (b) and (c) show the correspondence between the given EMB insertion depths and the computed tip positions at a $\mu-$scale.

The resulting tip micro-motion is shown in Fig. 4.5(b) for the naïve kinematic model (i.e. $\lambda = 0$). Figure 4.5(c) shows the tip motion for an updated model assuming $\lambda = 0.2 + 0.06q_s$. We note that, as expected, in both cases the robot straightens with EMB insertion since the robot body straightens. However, the updated model exhibits a turning point behavior which relates to the combined effect of straightening and change in the end effector angle θ'. This same phenomenon was observed experimentally in section 4.7.

4.6.2 Instantaneous Kinematics and Error Propagation

To verify the derivations of instantaneous kinematics and error propagation, we compute Jacobians following section 4.4.2. Since the simulation case represents the robot motion within a bending plane, the columns of the Jacobians represent 2×1 vectors of induced velocities for unit change in the variables associated with each Jacobian. The following simulations verify the macro motion Jacobian $\mathbf{J}_M$, the micro motion Jacobian $\mathbf{J}_\mu$, and the identification Jacobian $\mathbf{J_k}$ by plotting the Jacobian columns. The verification is carried out visually by verifying that the Jacobian columns induce tip velocity tangent to the trajectory generated by direct kinematics. In addition, each Jacobian has been also verified numerically via finite difference computations.

To verify $\mathbf{J}_M$, the EMB insertion depth q_s was fixed and direct actuation of backbones was assumed. Sample tip positions along the trajectory were obtained via direct kinematics and the corresponding Jacobian $\mathbf{J}_M$ was computed. Figure 4.6 shows the simulation results.

These results verify that the computed $\mathbf{J}_M$ is tangent to the macro motion trajectory.

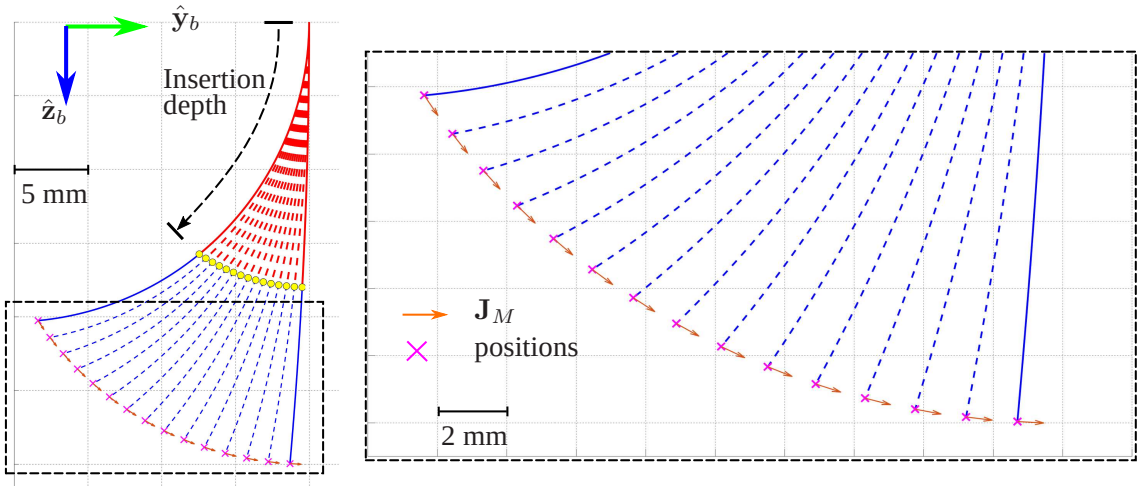
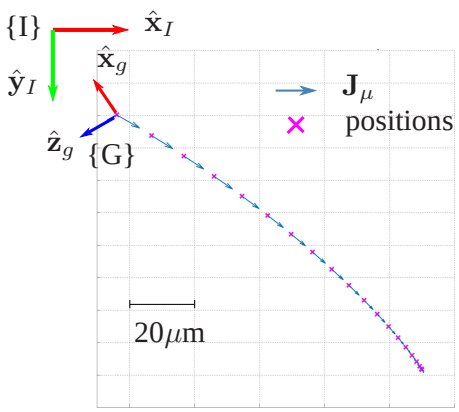


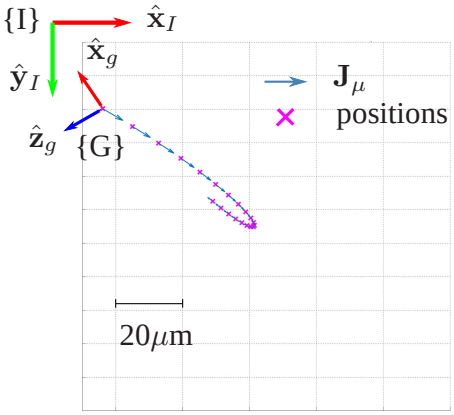
Figure 4.6: The macro motion simulation and the micro motion Jacobian computed during the simulation (red indicates inserted portion, blue indicates empty portion). The Jacobian $\mathbf{J}_M$ is shown in brown arrows representing induced tip velocities.

To verify $\mathbf{J}_\mu$, the secondary backbones were assumed locked and the EMB insertion depth q_s was varied. The Jacobian $\mathbf{J}_\mu$ was computed and plotted for each EMB depth. Two different scenarios of modeling uncertainty were considered: $\lambda = 0$ and $\lambda = 0.2 + 0.06q_s$. The results in Figure 4.7 (a) and (b) verify that $\mathbf{J}_\mu$ is tangent to the micro scale trajectory generated by direct kinematics.

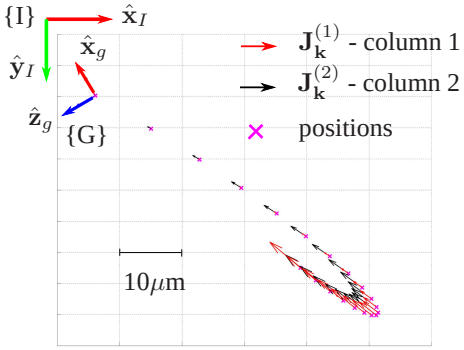
Figure 4.7(c) shows the plots of the identification Jacobian $\mathbf{J_k}$ for the simulation scenario where $\lambda \neq 0$, revealing how the parameter errors of modeling uncertainty affect the tip positions and hence the shape of the trajectory.



(a) micro motion Jacobian when $\lambda = 0$



(b) micro motion Jacobian when $\lambda \neq 0$



(c) error propagation using $\mathbf{J_k}$ and $\mathrm{d}\mathbf{k}_\lambda$ where perturbations were overlapped in simulated trajectory

Figure 4.7: Simulations verifying derivations of Jacobians.

4.7 Experimental Validations

In [21] the feasibility of micro motion through equilibrium modulation was demonstrated. The following experiments evaluate the ability of our simplified kinematic model to capture the micro-motion behavior, validate the calibration framework in section 4.5, and assess the accuracy of the updated kinematic model in reflecting the experimental data.

4.7.1 *Experimental Setup & Ground Truth Data*

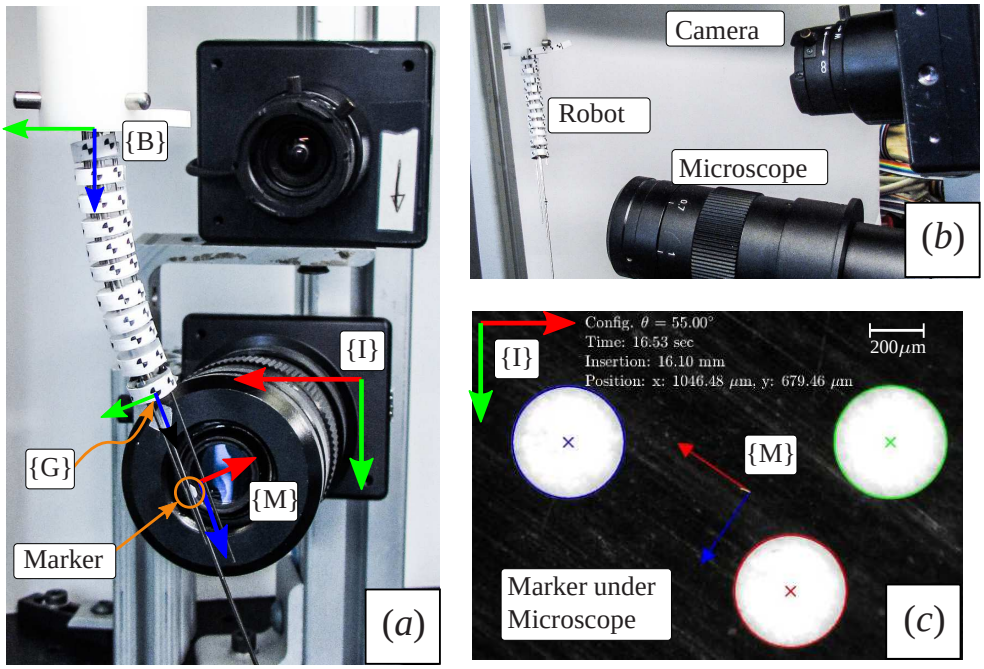


Figure 4.8: Experimental Setup: (a) a single-segment continuum robot whose motion is captured by two cameras; (b) the side view of the setup; (c) the segmented marker under the microscope view.

A single-segment continuum robot with EMB insertion actuation was used as the experimental platform, Figure 4.8. The platform was presented in [21], and it was modified from an earlier multi-backbone continuum robot design [102]. To observe the robot tip motion at different scales, one HD camera (FLIR Dragonfly II®) was used to capture the macro motion and the bending shape while an identical camera mounted on a 22.5× microscope lens to capture the micro motion. Custom "multi-circled" marker was used to track the tip motion under microscope while multiple custom "X" markers were attached to the continuum robot body to observe the bending shape. The vision measurement methods used were presented in [21] with the micro motion tracking accuracy being reported better than 2 μm.

Fig. 4.8 shows the frames used and also previously referred to in Fig. 4.5. The microscope is fixed at a known offset relative to the robot base, and such offset is represented as a

constant transformation from the image frame {I} to the robot base frame {B}. The tracked marker frame {M} is placed at a known offset relative to the end disk (gripper frame {G}), and the transformation is represented as a constant transformation between {M} and {G}. The marker position and orientation in the image frame is obtained by the segmentation of the three circles that construct an asymmetric pattern, as shown in Fig. 4.8(c).

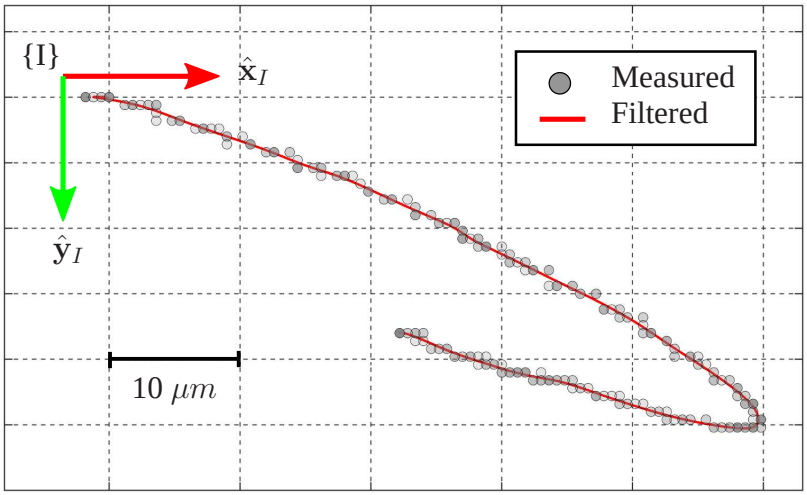


Figure 4.9: Image-segmented tip positions under microscope

Figure 4.9 shows a sample marker frame trajectory during EMB insertion. The marker positions were segmented from microscope images collected at 30 frames per second and were reported in camera frame {I}. Applying a butterworth infinite impulse response filter with the 3-dB frequency as 30 Hz, provided a smooth trajectory for calibration.

4.7.2 Model Calibration

Using calibration method in section 4.5, we calibrated the modeling uncertainty parameter vector $\mathbf{k}_\lambda$. The parameter vector $\mathbf{k}_\lambda$ in Eq. (4.18) consists of three elements, a bias term k_{λ_0}, a coefficient gain k_{λ_θ} that is associated with the nominal bending angle θ, and a coefficient gain k_{λ_q} that relates to the EMB insertion depth q_s. As a preliminary study, in this paper, we focus on investigating and calibrating k_{λ_0} and k_{λ_q}. Once the characterization of k_{λ_0} and k_{λ_q} is achieved, one can exhaust the choices of θ to investigate the effect of k_{λ_θ}.

Algorithm 4 was initialized with $k_{\lambda_0} = 0$, $k_{\lambda_q} = 0$, meaning no modeling uncertainty considered. In each iteration, the modeled positions were computed using the current estimates of the parameters. The *aggregated* error vector was then calculated between the modeled and experimental positions. For each iteration, both of the current estimates of the parameters and the position root-mean squared errors (RMSE) of all insertion samples (382 in total) were reported. A relative convergence threshold of 0.1% was used to determine the convergence.

For the particular experimental data collection, the parameter estimation (model calibration) went through 46 iterations before converging, where a step size of $\eta = 0.1$ was selected (η as introduced in Algorithm 4). Figure 4.10 shows selected iterations during the estimation, and the details of the iterations are reported in Table 4.3. The estimation started with an initial position RMSE of 44.27 μm, and after its convergence, the position RMSE was reduced to 5.82 μm, showing an improvement of 86.8% in model errors.

Table 4.3: Calibration using full micro motion trajectory

Iteration	0	5	10	20	30	45	46
$\frac{k_{\lambda_0}}{100}$	0	4.22	6.72	9.06	9.88	10.23	10.24
$\frac{k_{\lambda_q}}{1000}$	0	2.7	4.3	5.7	6.3	6.5	6.5
RMSE [μm]	**42.27**	15.71	25.40	7.72	6.07	5.82	**5.82**

Further investigating this calibration result, by dividing the tip trajectory into two segments, we observe that the current simplistic modeling approach produced bigger errors after the turning point: the RMSEs were reported as $4.87\mu m$ and $6.63\mu m$ for the two segments before and after turning point that had the lengths of $48.11\mu m$ and $38.82\mu m$, respectively. In practice, once the turning point location is identified, if one wishes to further improve the model accuracy, a model that only predicts the trajectory before the turning point may be considered. We therefore considered another calibration where only the partial micro motion trajectory before the turning point was used. With the same iteration step size

and convergence criterion, the estimation went through 59 iterations to converge, and the updated results were reported in Table 4.4 and plotted in Fig. 4.11. The position RMSE was then further improved to $4.76\mu m$.

Table 4.4: Calibration using partial micro motion trajectory

Iteration	0	5	10	20	30	58	59
$\frac{k_{\lambda_0}}{100}$	0	4.22	6.72	9.06	9.88	10.30	10.30
$\frac{k_{\lambda_q}}{1000}$	0	2.7	4.3	5.7	6.3	6.5	6.5
RMSE [μm]	**19.51**	13.01	9.30	6.12	5.17	4.76	**4.76**

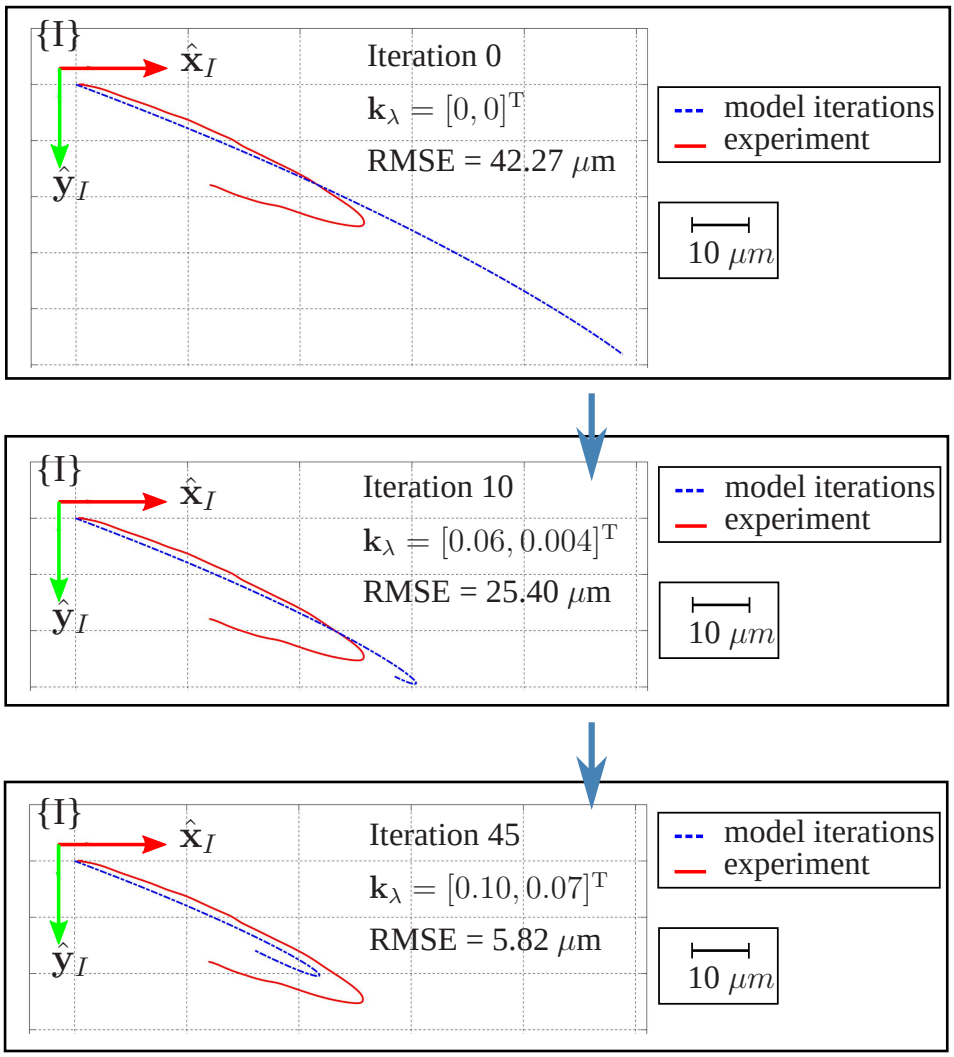


Figure 4.10: Experimental data and model iterations during the parameter estimation (model calibration) - using full micro motion trajectory.

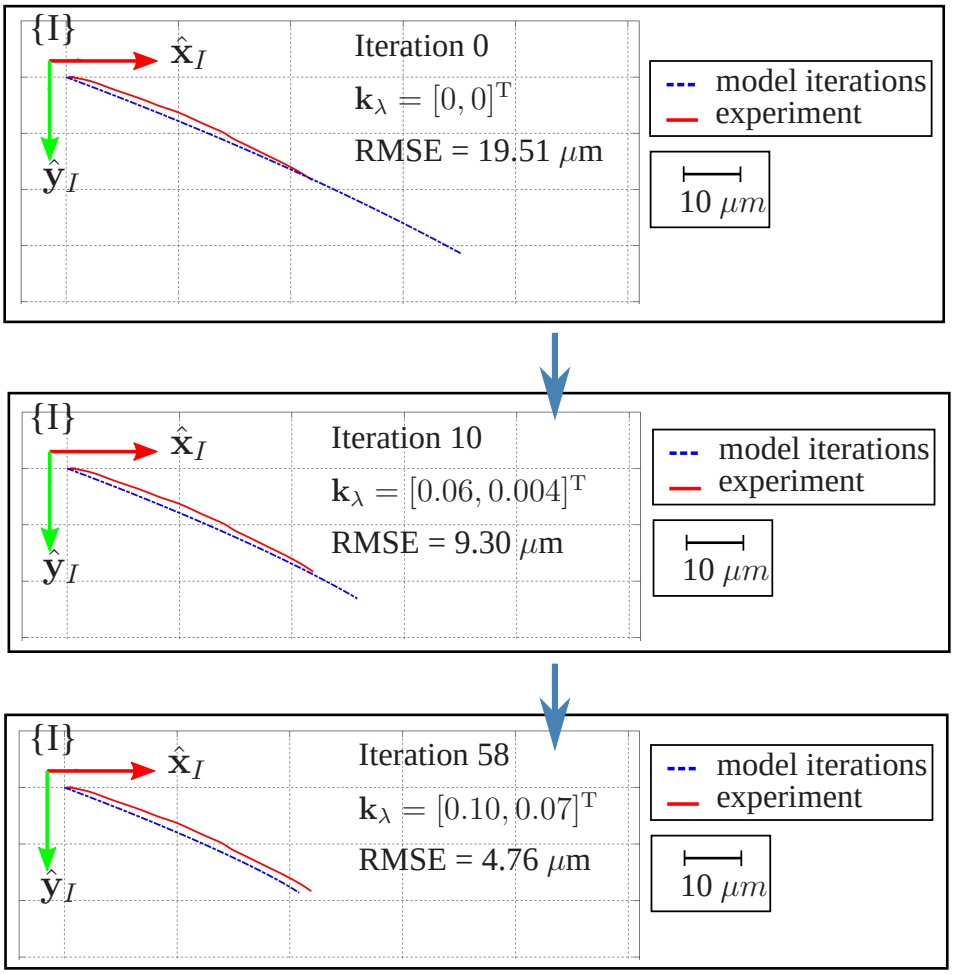


Figure 4.11: Experimental data and model iterations during the parameter estimation (model calibration) - using partial micro motion trajectory

4.7.3 Discussion

This work focused on creating a simplistic, yet fast model for equilibrium modulation control implementation. The kinematic model traded accurate mechanics modeling (which typically leads to solving nonlinear boundary value problems) with simplicity and speed of computation. Our experimental data showed an unexpected motion behavior manifested by a turning point along the micro-motion trajectory. The model presented in this paper does not offer a physical explanation to this behavior, but can capture this behavior for a given robot. The model calibration results indicate that there is still a potential to improve the

model performance by further investigating alternative modeling assumptions and different descriptions of modeling uncertainties. One of the limitations of our approach can be inferred from the simulation shown in Fig. 4.7, where both columns of the identification Jacobian are almost aligned with the tangent to the direct kinematics trajectory. The attainable correction directions that the column-space of $\mathbf{J_k}$ affords is therefore limited in reshaping the model trajectory. This was also observed from the experimental validation. Shown from the iterations in Fig. 4.10 and Fig. 4.11, it is difficult to reshape the modeled tip trajectory in the direction that is perpendicular to the trajectory. The other limitation is potentially caused by the choice of linearity in expressing the modeling uncertainty, which may not be descriptive enough.

4.8 Conclusion

This work presented the first modeling attempt for a new class of continuum robots capable of multi-scale motion. These robots achieve macro-scale and micro-scale motions through direct and indirect actuation (equilibrium modulation). Instead of focusing on a high-fidelity mechanics-based model, which typically leads to non-linear boundary value problems not easy to adopt for real-time control or parameter identification. Instead the paper presented a simplified mechanics-based model utilizing moment coupling effects between sub-segments of the continuum robot. This approach generates a differential kinematics model that covers both macro and micro-motion. As a result of unavoidable parameter uncertainty, we presented a model-calibration approach that can compensate for parameter inaccuracy, friction effects and modeling inaccuracies due to the simplistic modeling assumptions. The modeling approach along with the calibration framework was validated experimentally on a multi-backbone continuum robot. The calibrated model reported a positional root-mean-squared error as 5.83 μm if one wishes to use the model for the entire motion profile with the turning point. If one chooses to exclude motions past the turning point, the calibrated model fit the experimental data with an accuracy of 4.76

μm. Future work will focus on investigations on a more sophisticated models capable of incorporating geometric constraints as well as minimizing mechanical energy for improved model accuracy. In addition, effects of direction reversal of EMB insertion can manifest in hysteresis, which has not been explored in this work, but still remains the topic of ongoing research.

4.9 Relevant Published Work

Publications and manuscripts planned for future include:

J.1 **Wang, L.**, Del Giudice, G., and Simaan, N., "Simplified Kinematics of Continuum Robot Equilibrium Modulation via Moment Coupling Effects and Model Calibration", *submitted* to ***ASME Journal of Mechanisms and Robotics***, 2018, *under revision*.

C.1 Del Giudice, G., **Wang, L.**, Shen, J., Joos, K. and Simaan, N., "Continuum Robots for Multi-Scale Motion: Micro-Scale Motion Through Equilibrium Modulation", IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'), Vancouver, Canada, 2017, pp. 2537-2542.

Chapter 5

Conclusions

5.1 Summary of Findings

Continuum robots for surgical applications can support complex surgical tasks within deep confined spaces of the body. Such surgical paradigms often present surgeons with sensory and surgical scene interpretation challenges that diminish situational awareness. These robots can reach deep into the body and, in some scenarios, using them in a semi-automated mode of operation may alleviate the cognitive burden of the surgeons. Additionally, these robots have been recently shown to be able to support two modes of operation including macro-scale and micro-scale motion; thereby possibly enabling in-vivo microsurgical execution and image-based biopsy. These exciting capabilities of continuum robots are unattainable without the availability of accurate kinematic models of these robots. Furthermore, situational awareness augmentation requires methods for reconciling preoperative imaging information with the surgical scene in a way that helps the surgeon in executing surgical tasks safely.

The above-listed needs and opportunities guide the scope of this dissertation in two broad themes: 1) the augmentation of situational awareness through the use of force-controlled exploration as a means for updating the geometry of a virtual fixture despite possible organ shift/deformation, 2) an exploration of mathematical frameworks for modeling and calibration of continuum robots when moving in free space at a macro and micro-scale.

The key contributions of this dissertation lie in addressing the scientific gaps and application needs within each one of the two themes listed above. Current methods for updating the geometry of assistive telemanipulation virtual fixtures are predominantly limited to use of intraoperative imaging. In Chapter 2, we explore the utility of using force-controlled exploration as a means for updating the registration of an organ to a pre-operative model. Such methods may in the future be used as a means of augmenting other sources of information such as stereo-vision or organ laser scanning for the purpose of improving

registration. To achieve this goal, we present a model-mediated telemanipulation framework and force-controlled exploration for model update. Force-controlled ablation along a desired path with an assistive virtual fixture is demonstrated using the proposed telemanipulation framework. The method is validated on both a custom-made cartesian robot and a da Vinci Research Kit robot and the preliminary results suggest the feasibility of updating virtual fixtures based on force-controlled scans of organs.

In Chapter 3, we focus on presenting a mathematical framework for calibrating the kinematic model of continuum robots. An investigation of observability/sensitivity of kinematic mappings shows that direct calibration of the joint-to-task space mapping is not advisable for these robots. Instead, a two-staged approach where a mapping between joint and configuration space is calibrated and then the mapping from configuration space to task space is calibrated. The derivation of the identification Jacobians for calibration of continuum robots advances the literature, which has been limited to ad-hoc methods that work for limited architectures, but result in limited model update (e.g. look-up tables).

In Chapter 4, we investigate methods along the lines of extending calibration modeling to account for cases where continuum robots are used in micro-scale motion generation through equilibrium modulation. These robots present a new concept in continuum robotics, and we develop a simplified kinematic model based on moment coupling effect which lays the foundation to enabling future applications such as image-based biopsy and micro-surgery.

5.2 Future Research Directions

This dissertation complements previous theoretical works in continuum robots which supports them to be platforms with sensory and surgical intervention capabilities. Simaan et al. [87] proposed a novel design, the kinematics, and the statics of a multi-backbone continuum robot. Xu presented the intrinsic force sensing capabilities of multi-backbone continuum robots, along with the validity of the circular bending-shape assumption and

model-based actuation compensation. Goldman investigated algorithms for shape and stiffness estimation in compliant environments and a method for compliant motion of multi-backbone continuum robots. Bajo developed a collision detection algorithm that was paired with a constrained Extended Kalman filter for registering surgical continuum robots to compliant environments, and demonstrated hybrid motion/force control based stiffness imaging of compliant environments.

The method of force-controlled exploration for updating virtual fixture was demonstrated on continuum robots [106]. Future work will investigate methods of augmenting additional source of information to further improve the registration robustness and will develop frameworks of updating virtual fixture in real-time to support telemanipulation more responsively.

The calibration of continuum robots obtains an accurate kinematic model, enabling future work on semi-automated surgical tasks using such robots (e.g. ablation, suturing and needle insertion). The current geometric calibration framework (as presented in Chapter 3) assumes a prerequisite of obtaining an actuation compensation for the joint-to-configuration space mapping using the estimation methods in [77]. In the future, a unified framework that estimates the parameters of geometric calibration and actuation compensation in one step is more desirable. In addition, the current geometric calibration assumes non-forceful robot/environment interaction. Future work will investigate calibrating robots under forceful interaction condition considering the statics and equilibrium of the robots.

The simplified kinematics of continuum robot equilibrium modulation provides the first attempt of predicting and explaining the CREM micro motion. Future work will investigate more sophisticate and more generalized modeling approaches to further improve the accuracy, and will implement control strategies for image-based biopsy applications.

BIBLIOGRAPHY

[1] S. Abiko and K. Yoshida. On-line parameter identification of a payload handled by flexible based manipulator. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, volume 3, pages 2930–2935. IEEE, 2004.

[2] Jorge Angeles and Carlos S López-Cajún. *Optimization of cam mechanisms*, volume 9. Springer Science & Business Media, 2012.

[3] Peter Arbenz. Einführung in matlab (introduction to matlab).

[4] Elif Ayvali, Alexander Ansari, Long Wang, Nabil Simaan, and Howie Choset. Utility-guided palpation for locating tissue abnormalities. *IEEE Robotics and Automation Letters*, 2(2):864–871, 2017.

[5] Elif Ayvali, Rangaprasad Arun Srivatsan, Long Wang, Rajarshi Roy, Nabil Simaan, and Howie Choset. Using bayesian optimization to guide probing of a flexible environment for simultaneous registration and stiffness mapping. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 931–936. IEEE, 2016.

[6] Andrea Bajo, Roger E Goldman, Long Wang, Dennis Fowler, and Nabil Simaan. Integration and preliminary evaluation of an insertable robotic effectors platform for single port access surgery. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 3381–3387. IEEE, 2012.

[7] Andrea Bajo and Nabil Simaan. Finding lost wrenches: Using continuum robots for contact detection and estimation of contact location. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 3666–3673. IEEE, 2010.

[8] Andrea Bajo and Nabil Simaan. Kinematics-based detection and localization of contacts along multisegment continuum robots. *IEEE Transactions on Robotics*, 28(2):291–302, 2012.

[9] Andrea Bajo and Nabil Simaan. Hybrid motion/force control of multi-backbone continuum robots. *Int. J. Rob. Res.*, 35(4):422–434, April 2016.

[10] Seth D Billings, Emad M Boctor, and Russell H Taylor. Iterative most-likely point registration (imlp): a robust algorithm for computing optimal shape alignment. *PloS one*, 10(3), 2015.

[11] Jessica Burgner-Kahrs, D Caleb Rucker, and Howie Choset. Continuum robots for medical applications: A survey. *IEEE Transactions on Robotics*, 31(6):1261–1280, 2015.

[12] D. B. Camarillo, C. R. Carlson, and J. K. Salisbury. Configuration tracking for continuum manipulators with coupled tendon drive. *IEEE Transactions on Robotics*, 25(4):798–808, Aug 2009.

[13] Preetham Chalasani, Long Wang, Rajarshi Roy, Nabil Simaan, Russell H Taylor, and Marin Kobilarov. Concurrent nonparametric estimation of organ geometry and tissue stiffness using continuous adaptive palpation. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 4164–4171. IEEE, 2016.

[14] Preetham Chalasani, Long Wang, Rashid Yasin, Nabil Simaan, and Russell H Taylor. Preliminary evaluation of an online estimation method for organ geometry and tissue stiffness. *IEEE Robotics and Automation Letters*, 3(3):1816–1823, 2018.

[15] Chao Chen, Peifu Tang, Lihai Zhang, and Bin Wang. Treatment of fingertip degloving injury using the bilaterally innervated sensory cross-finger flap. *Annals of plastic surgery*, 73:645–51, 2014.

[16] Zihan Chen, Anton Deguet, Russel Taylor, Simon DiMaio, Gregory Fischer, and Peter Kazanzides. An open-source hardware and software platform for telesurgical robot research. In *MICCAI Workshop on Systems and Arch. for Computer Assisted Interventions*, Sep 2013.

[17] Gregory S Chirikjian. A general numerical method for hyper-redundant manipulator inverse kinematics. In *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on*, pages 107–112. IEEE, 1993.

[18] G.S. Chirikjian and J.W. Burdick. A modal approach to hyper-redundant manipulator kinematics. *IEEE Transactions on Robotics and Automation*, 10(3):343–354, June 1994.

[19] Changhyun Cho, Sungchul Kang, Munsang Kim, and Jae-Bok Song. Macro-micro manipulation with visual tracking and its application to wheel assembly. *International Journal of Control, Automation, and Systems*, 3(3):461–468, 2005.

[20] Mirko Daniele Comparetti, Alberto Vaccarella, Ilya Dyagilev, Moshe Shoham, Giancarlo Ferrigno, and Elena De Momi. Accurate multi-robot targeting for keyhole neurosurgery based on external sensor monitoring. *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*, 226(5):347–359, 2012.

[21] Giuseppe Del Giudice, Long Wang, Jin-Hui Shen, Karen Joos, and Nabil Simaan. Continuum robots for multi-scale motion: Micro-scale motion through equilibrium modulation. In *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*, pages 2537–2542. IEEE, 2017.

[22] Jienan Ding, Roger E Goldman, Kai Xu, Peter K. Allen, Dennis L. Fowler, and Nabil Simaan. Design and Coordination Kinematics of an Insertable Robotic Effectors Platform for Single-Port Access Surgery. *IEEE/ASME transactions on mechatronics*

: a joint publication of the IEEE Industrial Electronics Society and the ASME Dynamic Systems and Control Division, In press.:1612–1624, October 2013.

[23] Pierre E Dupont, Jesse Lock, Brandon Itkowitz, and Evan Butler. Design and control of concentric-tube robots. *Robotics, IEEE Transactions on*, 26(2):209–225, 2010.

[24] O. Egeland. Task-space tracking with redundant manipulators. *IEEE Journal on Robotics and Automation*, 3(5):471–475, October 1987.

[25] Vladimir Egorov, Heather Van Raalte, and Armen P Sarvazyan. Vaginal tactile imaging. *Biomedical Engineering, IEEE Transactions on*, 57(7):1736–1744, 2010.

[26] Konrad Entsfellner, Gero Strauss, Thomas Berger, Andreas Dietz, and Tim C. Lueth. Micro-Macro Telemanipulator for Middle-Ear Microsurgery. In *Robotics; Proceedings of ROBOTIK 2012; 7th German Conference on*, pages 1–4, Munich, Germany, 2012. VDE.

[27] Coralie Escande, Taha Chettibi, Rochdi Merzouki, Vincent Coelen, and Pushparaj Mani Pathak. Kinematic calibration of a multisection bionic manipulator. *IEEE/ASME Transactions on Mechatronics*, 20(2):663–674, 2015.

[28] Roy Featherstone, S.S. Thiebaut, and O. Khatib. A general contact model for dynamically-decoupled force/motion control. In *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, volume 4, pages 3281–3286. IEEE, 1999.

[29] James G Fujimoto. Optical coherence tomography for ultrahigh resolution in vivo imaging. *Nature biotechnology*, 21(11):1361–1367, 2003.

[30] AM Galea and RD Howe. Tissue stiffness from tactile imaging. In *Engineering in Medicine and Biology, 2002. 24th Annual Conference and the Annual Fall Meeting*

of the Biomedical Engineering Society EMBS/BMES Conference, 2002. Proceedings of the Second Joint, volume 2, pages 935–936. IEEE, 2002.

[31] Nicolo Garbin, Long Wang, James H Chandler, Keith L Obstein, Nabil Simaan, and Pietro Valdastri. A disposable continuum endoscope using piston-driven parallel bellow actuator. In *Medical Robotics (ISMR), 2018 International Symposium on*, pages 1–6. IEEE, 2018.

[32] Nicolo Garbin, Long Wang, James H Chandler, Keith L Obstein, Nabil Simaan, and Pietro Valdastri. Dual-continuum design approach for intuitive and ultra low-cost upper gastrointestinal endoscopy. *IEEE Transactions on Biomedical Engineering*, 2018, under review.

[33] Isuru S Godage, Gustavo A Medrano-Cerda, David T Branson, Emanuele Guglielmino, and Darwin G Caldwell. Modal kinematics for multisection continuum arms. *Bioinspiration & biomimetics*, 10(3):035002, 2015.

[34] Roger E Goldman, Andrea Bajo, Lara S MacLachlan, Ryan Pickens, S Duke Herrell, and Nabil Simaan. Design and performance evaluation of a minimally invasive telerobotic platform for transurethral surveillance and intervention. *IEEE transactions on biomedical engineering*, 60(4):918–925, 2013.

[35] Roger E Goldman, Andrea Bajo, and Nabil Simaan. Algorithms for autonomous exploration and estimation in compliant environments. *Robotica*, 31(1):71–87, 2013.

[36] Roger E Goldman, Andrea Bajo, and Nabil Simaan. Compliant motion control for multisegment continuum robots with actuation force sensing. *IEEE Transactions on Robotics*, 30(4):890–902, 2014.

[37] Ian Gravagne and I.D. Walker. On the kinematics of remotely-actuated continuum robots. In *Proceedings 2000 ICRA. Millennium Conference. IEEE*

International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065), volume 3, pages 2544–2550. IEEE, 2000.

[38] S Hirose, T Kado, and Y Umetani. Tensor Actutated Elastic Manipulator. In *Proceedings of the Sixth World Congress on Theory of Machines and Mechanisms*, pages 978–981, 1983.

[39] Agathe Hodac and Roland Y. Siegwart. Decoupled macro/micro-manipulator for fast and precise assembly operations: design and experiments. In Bradley J. Nelson and Jean-Marc Breguet, editors, *Proc. SPIE 3834, Microrobotics and Microassembly*, pages 122–130, August 1999.

[40] John M Hollerbach and Charles W Wampler. The calibration index and taxonomy for robot kinematic calibration methods. *The international journal of robotics research*, 15(6):573–591, 1996.

[41] Cheongjae Jang, Junhyoung Ha, Pierre E Dupont, and Frank Chongwoo Park. Toward on-line parameter estimation of concentric tube robots using a mechanics-based kinematic model. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pages 2400–2405. IEEE, 2016.

[42] Min Yang Jung, Anton Deguet, and Peter Kazanzides. A component-based architecture for flexible integration of robotic systems. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 6107–6112. IEEE, 2010.

[43] Takeharu Kanazawa, Shunji Sarukawa, Hirofumi Fukushima, Shoji Takeoda, Gen Kusaka, and Keiichi Ichimura. Current reconstructive techniques following head and neck cancer resection using microvascular surgery. *Annals of vascular diseases*, 4(3):189–95, January 2011.

[44] Ankur Kapoor, Ming Li, and R.H. Taylor. Constrained control for surgical assistant robots. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 231–236, May 2006.

[45] Yiannis Karayiannidis and Zoe Doulgeri. Adaptive control of robot contact tasks with on-line learning of planar surfaces. *Automatica*, 45(10):2374–2382, 2009.

[46] Peter Kazanzides, Zihan Chen, Anton Deguet, Gregory S. Fischer, Russell H. Taylor, and Simon DiMaio. An open-source research kit for the da Vinci® surgical robot. In *IEEE Intl. Conf. on Robotics and Auto. (ICRA)*, May 2014.

[47] Samuel B Kesner and Robert D Howe. Discriminating tissue stiffness with a haptic catheter: Feeling the inside of the beating heart. In *World Haptics Conference (WHC), 2011 IEEE*, pages 13–18. IEEE, 2011.

[48] Oussama Khatib. A Unified Approach for Motion and Force Control of Robot Manipulators: The Operational Space Formulation. *IEEE Journal of Robotics and Automation*, 3(1):43–53, 1987.

[49] Jungsik Kim, Farrokh Janabi-Sharifi, and Jung Kim. A physically-based haptic rendering for telemanipulation with visual information: Macro and micro applications. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3489–3494. IEEE, September 2008.

[50] Jochen Krücker, Sheng Xu, Neil Glossop, Anand Viswanathan, Jörn Borgert, Heinrich Schulz, and Bradford J Wood. Electromagnetic tracking for thermal ablation and biopsy guidance: clinical evaluation of spatial accuracy. *Journal of Vascular and Interventional Radiology*, 18(9):1141–1150, 2007.

[51] Michael DM Kutzer, Sean M Segreti, Christopher Y Brown, Mehran Armand, Russell H Taylor, and Simon C Mears. Design of a new cable-driven manipulator

with a large open lumen: Preliminary applications in the minimally-invasive removal of osteolysis. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 2913–2920. IEEE, 2011.

[52] David M Kwartowitz, S Duke Herrell, and Robert L Galloway. Toward image-guided robotic surgery: determining intrinsic accuracy of the da vinci robot. *International Journal of Computer Assisted Radiology and Surgery*, 1(3):157–165, 2006.

[53] David M. Kwartowitz, S. Duke Herrell, and Robert L. Galloway. Toward image-guided robotic surgery: determining intrinsic accuracy of the da Vinci robot. *International Journal of Computer Assisted Radiology and Surgery*, 1(3):157–165, October 2006.

[54] David M. Kwartowitz, S. Duke Herrell, and Robert L. Galloway. Update: Toward image-guided robotic surgery: determining the intrinsic accuracy of the daVinci-S robot. *International Journal of Computer Assisted Radiology and Surgery*, 1(5):301–304, February 2007.

[55] Charles L Lawson and Richard J Hanson. *Solving least squares problems*. SIAM, 1995.

[56] Ming Li, Masaru Ishii, and Russell H Taylor. Spatial motion constraints using virtual fixtures generated by anatomy. *IEEE Transactions on Robotics*, 23(1):4–19, 2007.

[57] Zheng Li, Hongliang Ren, Philip Wai Yan Chiu, Ruxu Du, and Haoyong Yu. A novel constrained wire-driven flexible mechanism and its kinematic analysis. *Mechanism and Machine Theory*, 95:59–75, 2016.

[58] Cristian A Linte, John Moore, and Terry M Peters. How accurate is accurate enough? a brief overview on accuracy considerations in image-guided cardiac interventions.

In *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*, pages 2313–2316. IEEE, 2010.

[59] Hongbin Liu, David P Noonan, Benjamin J Challacombe, Prokar Dasgupta, Lakmal D Seneviratne, and Kaspar Althoefer. Rolling mechanical imaging for tissue abnormality localization during minimally invasive surgery. *Biomedical Engineering, IEEE Transactions on*, 57(2):404–414, 2010.

[60] Joern Andreas Lohmeyer, Yasmin Kern, Daniel Schmauss, Felix Paprottka, Felix Stang, Frank Siemers, Peter Mailaender, and Hans-Guenther Machens. Prospective clinical study on digital nerve repair with collagen nerve conduits and review of literature. *Journal of reconstructive microsurgery*, 30:227–34, 2014.

[61] Wei Luo, Freddy T Nguyen, Adam M Zysk, Tyler S Ralston, John Brockenbrough, Daniel L Marks, Amy L Oldenburg, and Stephen A Boppart. Optical biopsy of lymph node morphology using optical coherence tomography. *Technology in cancer research & treatment*, 4(5):539–547, 2005.

[62] Tobias Mahl, Alexander Hildebrandt, and Oliver Sawodny. A variable curvature continuum kinematics for kinematic control of the bionic handling assistant. *IEEE Transactions on Robotics*, 30(4):935–949, 2014.

[63] Benjamin W Mooring, Zvi S Roth, and Morris R Driels. Fundamentals of manipulator calibration. 1991.

[64] Ryan J Murphy, Michael DM Kutzer, Sean M Segreti, Blake C Lucas, and Mehran Armand. Design and kinematic characterization of a surgical manipulator with a focus on treating osteolysis. *Robotica*, 32(06):835–850, 2014.

[65] Andriy Myronenko and Xubo Song. Point set registration: coherent point drift. *IEEE transactions on pattern analysis and machine intelligence*, 32(12):2262–75, December 2010.

[66] Yuki Nagatsu and Seiichiro Katsura. Macro-micro bilateral control using Kalman filter based state observer for noise reduction and decoupling of modal space. In *IECON 2013 - 39th Annual Conference of the IEEE Industrial Electronics Society*, pages 4192–4197. IEEE, November 2013.

[67] Ali Nahvi and JM Hollerbach. The noise amplification index for optimal pose selection in robot calibration. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 1, pages 647–654. IEEE, 1996.

[68] Mehrzad Namvar and Farhad Aghili. Adaptive force-motion control of coordinated robots interacting with geometrically unknown environments. *IEEE Transactions on Robotics*, 21(4):678–694, 2005.

[69] Maria Neumann and Jessica Burgner-Kahrs. Considerations for follow-the-leader motion of extensible tendon-driven continuum robots. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 917–923. IEEE, 2016.

[70] Kirk Nichols, Allison M Okamura, et al. Autonomous robotic palpation: Machine learning techniques to identify hard inclusions in soft tissues. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 4384–4389. IEEE, 2013.

[71] Andrew L Orekhov, Caroline B Black, John Till, Scotty Chung, and D Caleb Rucker. Analysis and validation of a teleoperated surgical parallel continuum manipulator. *IEEE Robotics and Automation Letters*, 1(2):828–835, 2016.

[72] Richard P Paul. *Robot manipulators: mathematics, programming, and control: the computer control of robot manipulators*. Richard Paul, 1981.

[73] Vladimir T. Portman, Ben-Zion Sandler, and Eliahu Zahavi. Rigid 6-DOF parallel platform for precision 3-D micromanipulation. *International Journal of Machine Tools and Manufacture*, 41(9):1229–1250, July 2001.

[74] G Robinson and J.B.C. Davies. Continuum robots - a state of the art. In *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, volume 4, pages 2849–2854. IEEE, 1999.

[75] William S. Rone and Pinhas Ben-Tzvi. Continuum Manipulator Statics Based on the Principle of Virtual Work. In *Volume 4: Dynamics, Control and Uncertainty, Parts A and B*, page 321. ASME, November 2012.

[76] William S Rone and Pinhas Ben-Tzvi. Continuum robot dynamics utilizing the principle of virtual power. *IEEE Transactions on Robotics*, 30(1):275–287, 2014.

[77] R. Roy, L. Wang, and N. Simaan. Modeling and estimation of friction, extension, and coupling effects in multisegment continuum robots. *IEEE/ASME Transactions on Mechatronics*, 22(2):909–920, April 2017.

[78] D. Caleb Rucker and Robert J. Webster III. Statics and Dynamics of Continuum Robots With General Tendon Routing and External Loading. *IEEE Transactions on Robotics*, 27(6):1033–1044, December 2011.

[79] D Caleb Rucker and Robert J Webster III. Statics and dynamics of continuum robots with general tendon routing and external loading. *IEEE Transactions on Robotics*, 27(6):1033–1044, 2011.

[80] Changhai Rul, Xihua Wang, and Shuxiang Guo. A Novel Tool Using SMA Actuator for cell puncturing. In *SICE Annual Conference 2007*, pages 254–258. IEEE, September 2007.

[81] Siddharth Sanan, Stephen Tully, Andrea Bajo, Nabil Simaan, and Howie Choset. Simultaneous compliance and registration estimation for robotic surgery. In *Proceedings of Robotics: Science and Systems*, Berkeley, USA, July 2014.

[82] K Schröer. Theory of kinematic modelling and numerical procedures for robot calibration. In Rolf Bernhardt, R Bernhardt, and S Albright, editors, *Robot calibration*. Springer Science & Business Media, 1993.

[83] Siddarth Sen, Animesh Garg, David V Gealy, Stephen McKinley, Yiming Jen, and Ken Goldberg. Automating multi-throw multilateral surgical suturing with a mechanical needle guide and sequential convex optimization. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 4178–4185. IEEE, 2016.

[84] Moshe Shoham. Twisting wire actuator. *Journal of Mechanical Design*, 127(3):441–445, 2005.

[85] Bruno Siciliano and Oussama Khatib. *Springer handbook of robotics*. Springer Science & Business Media, 2008.

[86] N. Simaan, R. Taylor, and P. Flint. A dexterous system for laryngeal surgery. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, volume 1, 2004.

[87] Nabil Simaan. Snake-Like Units Using Flexible Backbones and Actuation Redundancy for Enhanced Miniaturization. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 3012–3017, Barcelona, Spain, 2005. IEEE.

[88] Nabil Simaan, Russell Taylor, and Paul Flint. A dexterous system for laryngeal surgery. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, pages 351–357 Vol.1, New Orleans, 2004. IEEE.

[89] Nabil Simaan, Kai Xu, Wei Wei, Ankur Kapoor, P Kazanzides, P Flint, and R Taylor. Design and Integration of a Telerobotic System for Minimally Invasive Surgery of

the Throat. *International Journal of Robotics Research - special issue on Medical Robotics (special Issue on Medical Robotics)*, 28(9):1134–1153, 2009.

[90] David J Slutsky. The management of digital nerve injuries. *The Journal of hand surgery*, 39:1208–15, 2014.

[91] Rangaprasad Arun Srivatsan, Elif Ayvali, Long Wang, Rajarshi Roy, Nabil Simaan, and Howie Choset. Complementary model update: A method for simultaneous registration and stiffness mapping in flexible environments. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 924–930. IEEE, 2016.

[92] Stephen Tully, Andrea Bajo, George Kantor, Howie Choset, and Nabil Simaan. Constrained filtering with contact detection data for the localization and registration of continuum robots in flexible environments. In *2012 IEEE International Conference on Robotics and Automation*, pages 3388–3394. IEEE, May 2012.

[93] Ian D Walker. Continuous backbone "continuum" robot manipulators. *ISRN Robotics*, 2013, 2013.

[94] Long Wang, Giuseppe Del Giudice, and Nabil Simaan. Simplified kinematics of continuum robot equilibrium modulation via moment coupling effects and model calibration. *Journal of Mechanisms and Robotics*, 2018, under review.

[95] Long Wang and Nabil Simaan. Investigation of error propagation in multi-backbone continuum robots. In *Advances in Robot Kinematics*, pages 385–394. Springer, 2014.

[96] Long Wang and Nabil Simaan. Investigation of Error Propagation in Multi-Backbone Continuum Robots. In *Advances in Robot Kinematics*, pages 385–394, 2014.

[97] Long Wang and Nabil Simaan. Geometric calibration of continuum robots: Joint space and equilibrium shape deviations. *IEEE Transactions on Robotics*, 2018, revision under review.

[98] Robert J Webster and Bryan A Jones. Design and kinematic modeling of constant curvature continuum robots: A review. *The International Journal of Robotics Research*, 2010.

[99] Robert J Webster, Joseph M Romano, and Noah J Cowan. Kinematics and calibration of active cannulas. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 3888–3895. IEEE, 2008.

[100] Robert J Webster III, Joseph M Romano, and Noah J Cowan. Mechanics of precurved-tube continuum robots. *IEEE Transactions on Robotics*, 25(1):67–78, 2009.

[101] Daniel E Whitney. Resolved motion rate control of manipulators and human prostheses. *IEEE Transactions on man-machine systems*, 1969.

[102] Kai Xu and Nabil Simaan. An Investigation of the Intrinsic Force Sensing Capabilities of Continuum Robots. *IEEE Transactions on Robotics*, 24(3):576–587, June 2008.

[103] Kai Xu and Nabil Simaan. Analytic Formulation for Kinematics, Statics and Shape Restoration of Multibackbone Continuum Robots via Elliptic Integrals. *ASME Journal of Mechanisms and Robotics (JMR)*, 2:11006–11013, 2010.

[104] Kai Xu and Nabil Simaan. Analytic Formulation for Kinematics, Statics and Shape Restoration of Multibackbone Continuum Robots via Elliptic Integrals. *Journal of Mechanisms and Robotics*, 2:011006, 2010.

[105] Kai Xu and Nabil Simaan. Intrinsic wrench estimation and its performance index for multisegment continuum robots. *Robotics, IEEE Transactions on*, 26(3):555–561, 2010.

[106] Rashid Yasin, Long Wang, Colette Abah, and Nabil Simaan. Using continuum robots for force-controlled semi autonomous organ exploration and registration. In *Medical Robotics (ISMR), 2018 International Symposium on*, pages 1–6. IEEE, 2018.

[107] Yuehong Yin, Hui Hu, and Yanchun Xia. Active tracking of unknown surface using force sensing and control technique for robot. *Sensors and Actuators A: Physical*, 112(2):313–319, 2004.

[108] Tsuneo Yoshikawa and Akio Sudou. Dynamic hybrid position/force control of robot manipulators-on-line estimation of unknown constraint. *IEEE Transactions on Robotics and Automation*, 9(2):220–226, 1993.

[109] Haoran Yu, Jin-Hui Shen, Karen M Joos, and Nabil Simaan. Calibration and integration of b-mode optical coherence tomography for assistive control in robotic microsurgery. *Transactions on Mechatronics*, 21(6), 2016.

[110] Yuan Yun and Yangmin Li. A novel design and analysis of a 3-DOF parallel manipulator for micro/nano manipulation. In *2008 IEEE Workshop on Advanced robotics and Its Social Impacts*, pages 1–6. IEEE, August 2008.

[111] Kourosh E Zanganeh and Jorge Angeles. The inverse kinematics of hyper-redundant manipulators using splines. In *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on*, volume 3, pages 2797–2802. IEEE, 1995.

[112] Jian Zhang and Nabil Simaan. Design of underactuated steerable electrode arrays for optimal insertions. *Journal of Mechanisms and Robotics*, 5(1):011008, 2013.

[113] Siyang Zuo, Michael Hughes, and Guang-Zhong Yang. Flexible robotic scanning device for intraoperative endomicroscopy in mis. *IEEE/ASME Transactions on Mechatronics*, 22(4):1728–1735, 2017.

Appendix A

Differential of a Homogeneous Transformation

The topic of kinematic error model for robot calibration was addressed in chapters 2.4 and 2.5 of Mooring [63]. This model relates small error in nominal manipulator kinematics to end effector (EE) position and orientation errors and this approach was illustrated using a standard Denavit Hartenberg kinematic model representation.

The final goal of this strategy is to achieve a differential model describing the relationship between changes in the calibration parameters and the end-effector pose as described by DH direct kinematics parametrization. Let $\mathbf{T} \in \mathbb{R}^{4\times4}$ be the homogenous transformation describing the end effector pose. Let $\mathbf{k}$ be the vector of calibration parameters (this would be the vector containing the DH parameters if we use DH parametrization). The complete differential of $\mathbf{T}$ is given by 2.43 in [63]:

$$\mathrm{d}\mathbf{T} = \sum_{\mathrm{i}=1}^{\mathrm{p}} \frac{\partial \mathbf{T}}{\partial \mathrm{k_i}} \Delta \mathrm{k_i} = \mathbf{T}\delta\mathbf{T}, \quad \mathbf{k} = (\mathrm{k_1}, \ldots, \mathrm{k_p})^{\mathrm{T}} \tag{A.1}$$

The matrix $\mathrm{d}\mathbf{T} \in \mathbb{R}^{4\times4}$ is the total differential of $\mathbf{T}$. The matrix $\delta\mathbf{T} \in \mathbb{R}^{4\times4}$ is the *differential* transformation matrix which only contains first order variations of the kinematic parameters. Here are some observations about equation (A.1):

(a) The derivative of a transformation can be written in a form of itself post-multiplied with a special matrix $\delta\mathbf{T}$;

(b) This $\delta\mathbf{T}$ is in $se(3)$ group, and it is constructed by a 3×3 skew symmetric matrix and a 3×1 vector; this result was earlier obtained by Richard Paul (equation 4.28 in [72]).

(c) Later this special matrix $\delta\mathbf{T}$ will be partitioned to define the translational and rotational errors on end-effector error at configuration $\mathbf{T}$.

As illustrated in [63], using the DH representation of link transformation as an example and denoting the DH parameters $\theta_n, \alpha_n, r_n, l_n$ for link n, the DH transformation $\mathbf{A}_n$ is given

by equation (A.2) (2.12 in [63]).

$$\mathbf{A}_n = \begin{bmatrix} \cos\theta_n & -\sin\theta_n\cos\alpha_n & \sin\theta_n\sin\alpha_n & l_n\cos\theta_n \\ \sin\theta_n & \cos\theta_n\cos\alpha_n & -\cos\theta_n\sin\alpha_n & l_n\sin\theta_n \\ 0 & \sin\alpha_n & \cos\alpha_n & r_n \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.2}$$

In the following paragraphs we solve for $\delta\mathbf{A}_n$. First let us derive the derivative $\mathrm{d}\mathbf{A}_\mathrm{n}$ as below (equation (2.16) in [63]):

$$\mathrm{d}\mathbf{A}_\mathrm{n} = \frac{\partial\mathbf{A}_\mathrm{n}}{\partial\theta_\mathrm{n}}\Delta\theta_\mathrm{n} + \frac{\partial\mathbf{A}_\mathrm{n}}{\partial\alpha_\mathrm{n}}\Delta\alpha_\mathrm{n} + \frac{\partial\mathbf{A}_\mathrm{n}}{\partial\mathrm{r}_\mathrm{n}}\Delta\mathrm{r}_\mathrm{n} + \frac{\partial\mathbf{A}_\mathrm{n}}{\partial\mathrm{l}_\mathrm{n}}\Delta\mathrm{l}_\mathrm{n} \tag{A.3}$$

Then $\mathrm{d}\mathbf{A}_\mathrm{n}$ may be rewritten or decomposed in a format of $\mathbf{A}_n\delta\mathbf{A}_n$. This step is not trivial and this particular example is done in chapter 2.4 in [63]. The result of $\delta\mathbf{A}_n$ is obtained as below (2.30 in [63]),

$$\delta\mathbf{A}_n = \begin{bmatrix} 0 & -c_{\alpha_n}\Delta\theta_n & s_{\alpha_n}\Delta\theta_n & \Delta l_n \\ c_{\alpha_n}\Delta\theta_n & 0 & -\Delta\alpha_n & l_n c_{\alpha_n}\Delta\theta_n + s_{\alpha_n}\Delta r_n \\ -s_{\alpha_n}\Delta\theta_n & \Delta\alpha_n & 0 & -l_n s_{\alpha_n}\Delta\theta_n + c_{\alpha_n}\Delta r_n \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{A.4}$$

Let us verify the result in this example: the variation of the first element $\mathbf{A}_{n_{1,1}}$ is $\delta(\cos(\theta_n)) = -\sin(\theta_n)\delta\theta_n$. This same result is obtained by multiplying the first row of $\mathbf{A}_n$ with the first column of $\delta\mathbf{A}_n$. This example demonstrates that by introducing small errors in the joint and parameters $(\Delta\theta_n, \Delta\alpha_n, \Delta r_n, \Delta l_n)$, we can propagate the resulting pose error of the end-effector $\delta\mathbf{A}_n$.

We will henceforth use $\Delta\mathbf{X}$ to denote the vector representation of the differential $\delta\mathbf{X}$ for matrix $\mathbf{X} \in \mathbb{R}^{4\times4}$. Because of the skew symmetry of the top left 3×3 sub-matrix of $\delta\mathbf{T}$ (or $\delta\mathbf{A}_n$), we can partition this 4×4 matrix into two 3×1 vectors which are defined in

equation (2.49) in [63].

$$\Delta\mathbf{T} \triangleq \begin{bmatrix} \mathbf{d} \\ \boldsymbol{\delta} \end{bmatrix} \tag{A.5}$$

If we apply the notation of equation (A.5) in the above example, we have the following results as presented in (2.31, 2.32) of [63],

$$\begin{cases} \mathbf{d}_n = \begin{bmatrix} \Delta l_n \\ l_n c_{\alpha_n} \Delta\theta_n + s_{\alpha_n} \Delta r_n \\ -l_n s_{\alpha_n} \Delta\theta_n + c_{\alpha_n} \Delta r_n \end{bmatrix} = \begin{bmatrix} 0 \\ l_n c_{\alpha_n} \\ -l_n s_{\alpha_n} \end{bmatrix} \Delta\theta_n + \begin{bmatrix} 0 \\ s_{\alpha_n} \\ c_{\alpha_n} \end{bmatrix} \Delta r_n + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \Delta l_n \\ \boldsymbol{\delta}_n = \begin{bmatrix} \Delta\alpha_n \\ s_{\alpha_n} \Delta\theta_n \\ c_{\alpha_n} \Delta\theta_n \end{bmatrix} = \begin{bmatrix} 0 \\ s_{\alpha_n} \\ c_{\alpha_n} \end{bmatrix} \Delta\theta_n + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \Delta\alpha_n \end{cases} \tag{A.6}$$

By defining the following vectors

$$\boldsymbol{\rho}_n^1 = [0, l_n c_{\alpha_n}, -l_n s_{\alpha_n}]^T, \boldsymbol{\rho}_n^2 = [0, s_{\alpha_n}, c_{\alpha_n}]^T, \boldsymbol{\rho}_n^3 = [1, 0, 0]^T \tag{A.7}$$

We can rewrite equation (A.6) as the result for kinematic error model for single link DH representation:

$$\Delta\mathbf{A}_n \triangleq \begin{bmatrix} \mathbf{d}_n \\ \boldsymbol{\delta}_n \end{bmatrix} = \begin{bmatrix} \boldsymbol{\rho}_n^1 \Delta\theta_n + \boldsymbol{\rho}_n^2 \Delta r_n + \boldsymbol{\rho}_n^3 \Delta l_n \\ \boldsymbol{\rho}_n^2 \Delta\theta_n + \boldsymbol{\rho}_n^3 \Delta\alpha_n \end{bmatrix} \tag{A.8}$$

With the vector representation of equation (A.5), equation (A.1) can be formulated as a conventional geometric Jacobian as in (2.57)

$$\Delta\mathbf{T} = \mathbf{J}_K \Delta\mathbf{k} \tag{A.9}$$

Observations on equation (A.9):

(a) if we separate the joint variables off the parameter vector $\mathbf{k}$, we obtain a more rigorous treatment of $\Delta\mathbf{T}$. Then we call this particular conventional Jacobian $\mathbf{J_k}$ the *Identification* Jacobian. This definition is given in (4.6) in [63].

(b) $\Delta\mathbf{T}$ is defined as the *differential translation and rotation vector*

For the example we started in this section, we can formulate the Jacobian using the vectors defined in equation (A.7),

$$\Delta\mathbf{T} = \mathbf{J}_K \delta\mathbf{k} = \begin{bmatrix} \boldsymbol{\rho}_n^1 & \boldsymbol{\rho}_n^2 & \boldsymbol{\rho}_n^3 & 0 \\ \boldsymbol{\rho}_n^2 & 0 & 0 & \boldsymbol{\rho}_n^3 \end{bmatrix} \delta\mathbf{k} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ l_n c_{\alpha_n} & s_{\alpha_n} & 0 & 0 \\ -l_n s_{\alpha_n} & c_{\alpha_n} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ s_{\alpha_n} & 0 & 0 & 0 \\ c_{\alpha_n} & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta\theta_n \\ \Delta r_n \\ \Delta l_n \\ \Delta\alpha_n \end{bmatrix} \tag{A.10}$$

Equation (A.10) expressed the kinematic error model which relates the joint and parameter error to EE error. Notice that the example only formulated one link. We thereby summarize the definition of kinematic error model as the following.

Given a 4×4 transformation matrix $\mathbf{T}$, its *first-order* Taylor expansion $\mathrm{d}\mathbf{T}$ can be decomposed as the following,

$$\mathrm{d}\mathbf{T} = \sum_{\mathrm{i}=1}^{\mathrm{p}} \frac{\partial \mathbf{T}}{\partial \mathrm{k_i}} \Delta \mathrm{k_i} = \mathbf{T}\delta\mathbf{T}, \quad \mathbf{k} = (\mathrm{k}_1, \ldots, \mathrm{k_p})^{\mathrm{T}} \tag{A.11}$$

where $\delta\mathbf{T} \in se(3)$ and it is partitioned into into two 3×1 vectors which are defined as *differential translation and rotation vectors*.

$$\delta\mathbf{T}_{4\times4} \Rightarrow \Delta\mathbf{T}_{6\times1} = \begin{bmatrix} \mathbf{d} \\ \boldsymbol{\delta} \end{bmatrix} \tag{A.12}$$

Then writing (A.12) in Jacobian format, results in:

$$\Delta \mathbf{T} = \mathbf{J_k} \Delta \mathbf{k} \tag{A.13}$$

Without joint variables in $\mathbf{k}$, $\mathbf{J_k}$ is defined as the *Identification Jacobian*

The definition of error propagation above is important and it is used as the preferable definition of *Identification Jacobian*. Another popular *Identification Jacobian* definition associated with the use of loop closure equations is used in Nahvi & Hollerbach [67] and Hollerbach & Wampler [40]:

$$\mathbf{f}^i = \mathbf{g}^i(\mathbf{k}, \boldsymbol{\eta}^i) \approx \mathbf{0} \tag{A.14}$$

where $\mathbf{f}^i$ is a residual function of i^{th} pose loop closure equation, $\mathbf{g}^i$ is the loop closure equation, $\mathbf{x}$ is a vector of robot parameters to be calibrated and $\boldsymbol{\eta}^i$ is the joint transducer reading.

Combining equation (A.14) for all end effector poses results in a vector representation:

$$\mathbf{f} = \mathbf{g}(\mathbf{k}, \boldsymbol{\eta}) \tag{A.15}$$

where $\mathbf{f} = [\mathbf{f}^{1^T} \dots \mathbf{f}^{m^T}]^T$ and $\mathbf{g} = [\mathbf{g}^{1^T} \dots \mathbf{g}^{m^T}]^T$. Linearizing equation (A.15),

$$\Delta \mathbf{f} = \frac{\partial \mathbf{g}}{\partial \mathbf{x}} \Delta \mathbf{k} = \mathbf{C} \Delta \mathbf{k} \tag{A.16}$$

where is $\Delta \mathbf{f}$ the error between measured and computed residual function, $\mathbf{C}$ is the identification Jacobian, and $\Delta \mathbf{x}$ is the correction to be applied to the current parameter estimate.

Appendix B

Differential Kinematics for Geometric Calibration of Multi-backbone Continuum Robots

B.1 Deriving $\mathbf{J}_{\mathbf{x}t}$ and $\mathbf{J}_{\mathbf{x}\delta}$

The geometric Jacobian, $\mathbf{J}_{\mathbf{x}_p t}$, that was first introduced in equation (3.15), relates the infinitesimal (δt) to the in-plane kinematic variable $(\delta \mathbf{x}_p)$, and is derived as below:

$$\frac{\partial^1 \mathbf{p}_{xz}}{\partial t} = \int_0^l \left(\begin{bmatrix} -\sin\theta_s(s) \\ \cos\theta_s(s) \end{bmatrix} \frac{\partial}{\partial t}\theta_s(s) \right) \mathrm{ds} \tag{B.1}$$

$$\frac{\partial}{\partial t}\theta_s(s) = \int_0^s \left[\kappa_b(\tau) - \kappa_a(\tau)\right] \mathrm{d}\tau \tag{B.2}$$

For the purpose of completeness, another geometric Jacobian, $\mathbf{J}_{\mathbf{x}t}$, that relates the infinitesimal (δt) to the instantaneous twist $\delta\boldsymbol{\xi} \in \mathbb{R}^6$, is also derived here following similar methodology as in 3.3.2. First, we partition the translation and rotation partitions as:

$${}^b\delta\boldsymbol{\xi}(\delta t) \triangleq \left[({}^b\delta\mathbf{x})^{\mathrm{T}}, ({}^b\delta\boldsymbol{\mu})^{\mathrm{T}} \right]^{\mathrm{T}}, \quad {}^b\delta\boldsymbol{\xi} \in \mathbb{R}^{6\times 1} \tag{B.3}$$

The segment-end translational and angular velocities are both obtained using chain rule:

$${}^b\delta\mathbf{x}_\kappa = {}^b\mathbf{R}_1 \mathbf{S}_\mathbf{x}\, \delta\mathbf{x}_p, \quad \mathbf{S}_\mathbf{x} \triangleq \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad {}^b\mathbf{R}_1 = e^{-\delta_0 [\mathbf{z}^\wedge]} \tag{B.4}$$

$${}^b\delta\boldsymbol{\mu}_\kappa = -\hat{\mathbf{y}}_1 \mathbf{S}_{\boldsymbol{\mu}} \delta\mathbf{x}_p, \quad \mathbf{S}_{\boldsymbol{\mu}} \triangleq \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \tag{B.5}$$

Then the geometric Jacobian $\mathbf{J}_{\mathbf{x}t}$ is given as:

$${}^b\delta\boldsymbol{\xi}(\delta t) = \left[\begin{array}{c} {}^b\mathbf{R}_1 \mathbf{S}_\mathbf{x} \mathbf{J}_{\mathbf{x}_p t} \\ \hdashline -\hat{\mathbf{y}}_1 \mathbf{S}_{\boldsymbol{\mu}} \mathbf{J}_{\mathbf{x}_p t} \end{array} \right] \delta t \quad \triangleq \mathbf{J}_{\mathbf{x}t} \delta t \tag{B.6}$$

B.2 Deriving $\mathbf{J}_{\mathbf{k}_\kappa}$

The bending shape identification Jacobian, $\mathbf{J}_{\mathbf{k}_\kappa}$, first introduced in equation (3.18) and then defined in equation (3.19), relates bending shape parameter errors $\delta\mathbf{k}_\kappa$ to in-plane task space errors $\delta\mathbf{x}_p$. The derivation is shown as follows:

$$\mathbf{J}_{\mathbf{k}_\kappa} = \frac{\partial \mathbf{f}_p(t,\mathbf{k}_\kappa)}{\partial \mathbf{k}_\kappa} = [\frac{\partial \mathbf{f}_p}{\partial l}, \frac{\partial \mathbf{f}_p}{\partial \mathbf{a}}, \frac{\partial \mathbf{f}_p}{\partial \mathbf{b}}] \tag{B.7}$$

$$\frac{\partial \mathbf{f}_p}{\partial l} = [c_{\theta_s}(l), s_{\theta_s}(l), \kappa(l)]^T \tag{B.8}$$

$$\frac{\partial \mathbf{f}_p}{\partial \mathbf{a}} = (1-t)\int_0^l \begin{bmatrix} [-s_{\theta_s}, c_{\theta_s}]^{\mathrm{T}}\, \boldsymbol{\nu}^{\mathrm{T}} \\ \boldsymbol{\eta}^{\mathrm{T}} \end{bmatrix} \mathrm{d}s \tag{B.9}$$

$$\frac{\partial \mathbf{f}_p}{\partial \mathbf{b}} = t\int_0^l \begin{bmatrix} [-s_{\theta_s}, c_{\theta_s}]^{T}\boldsymbol{\nu}^{\mathrm{T}} \\ \boldsymbol{\eta}^{\mathrm{T}} \end{bmatrix} \mathrm{d}s \tag{B.10}$$

$$\boldsymbol{\nu}(s) = \int_0^s \boldsymbol{\eta}(\tau)\mathrm{d}\tau = \begin{bmatrix} s & \\ & \frac{s}{2} \end{bmatrix}\boldsymbol{\eta}(s), \quad \boldsymbol{\eta}(s) = [1, s^1]^T \tag{B.11}$$

B.3 Derivation of Equation (3.27)

As shown in equation (3.26), to obtain the full derivative d$\mathbf{R}$ of the rotation matrix that represents the orientation of the end-effector, an intermediate derivative matrix, $\frac{\partial \mathbf{R}}{\partial \gamma(s)}$, is needed. The following shows the steps in obtaining the expression of $\frac{\partial \mathbf{R}}{\partial \gamma(s)}$, and the resulted expression is given in equation (3.27).

Using (3.12) and (3.7), we have

$$\mathbf{R}(\psi, s, \mathbf{k}_\kappa, \mathbf{k}_\gamma) = e^{-\delta_0[\mathbf{e}_3{}^\wedge]}\ e^{(\frac{\pi}{2}-\theta_e)[\mathbf{e}_2{}^\wedge]}\ e^{\delta(s)[\mathbf{e}_3{}^\wedge]} \tag{B.12}$$

$$\frac{\partial \mathbf{R}}{\partial \gamma(s)} = e^{-\delta_0[\mathbf{e}_3{}^\wedge]}\ e^{(\frac{\pi}{2}-\theta_s)[\mathbf{e}_2{}^\wedge]}\ \left(\frac{\partial}{\partial \gamma(s)} e^{\delta(s)[\mathbf{e}_3{}^\wedge]}\right) \tag{B.13}$$

Deriving the differentiation of $e^{\delta(s)[\mathbf{e}_3\wedge]}$ results in $\frac{\partial \mathbf{R}}{\partial \gamma(s)}$ as:

$$\begin{aligned}\frac{\partial}{\partial \gamma_e} e^{\delta(s)[\mathbf{e}_3\wedge]} &= \left(e^{\delta(s)[\mathbf{e}_3\wedge]}\right) \frac{\partial}{\partial \gamma_e}\delta(s)[\mathbf{e}_3\wedge] \\ &= \left(e^{\delta(s)[\mathbf{e}_3\wedge]}\right)[\mathbf{e}_3\wedge]\end{aligned} \tag{B.14}$$

$$\frac{\partial \mathbf{R}}{\partial \gamma(s)} = e^{-\delta_0[\mathbf{e}_3\wedge]}\ e^{(\frac{\pi}{2}-\theta_e)[\mathbf{e}_2\wedge]}\ e^{\delta(s)[\mathbf{e}_3\wedge]}\ \left[\mathbf{e}_3\wedge\right] \tag{B.15}$$

Appendix C

Algorithms & Simulations for Geometric Calibration Multi-backbone Continuum Robots

C.1 Deriving $\frac{\partial \mathbf{c}_{\gamma_j}}{\partial \mathbf{k}_\gamma}$

Recall from Eqs. (3.70, 4.61, 4.62) that $\mathbf{c}_{\gamma_j}$ is the vector representation of the rotation matrix $\mathbf{R}_e = \bar{\mathbf{R}}_j \mathbf{R}_j^{\mathrm{T}}$, meaning that $\Delta \mathbf{c}_{\gamma_j}$ may be derived from $\mathrm{d}\mathbf{R}_{\mathrm{e}}$. Following the same process as (3.28), we have:

$$\begin{aligned} \mathrm{d}\mathbf{R}_{\mathrm{e_j}} &= \bar{\mathbf{R}}_j \, \mathrm{d}\left(\mathbf{R}_j^{\mathrm{T}}\right) = \bar{\mathbf{R}}_j \mathbf{R}_j^{\mathrm{T}} \, \delta\left(\mathbf{R}_j^{\mathrm{T}}\right) \\ &= \mathbf{R}_{e_j} \, \delta\left(\mathbf{R}_j^{\mathrm{T}}\right) \end{aligned} \tag{C.1}$$

If using the process in Eq. (3.28) directly on $\mathbf{R}_e$, one may write $\mathrm{d}\mathbf{R}_e = \mathbf{R}_e \delta \mathbf{R}_e$, which leads to $\delta \mathbf{R}_e = \delta\left(\mathbf{R}_j^{\mathrm{T}}\right)$, or

$$\delta \mathbf{R}_e = -\delta \mathbf{R}_j \tag{C.2}$$

Note that $\delta \mathbf{c}_{\gamma_j}$ is a vector that represents the skew-symmetric matrix $\delta \mathbf{R}_e$ while $\delta \boldsymbol{\mu}_\gamma$ in Eq. (3.30) may represent $\delta \mathbf{R}_j$ in this case. Using Eqs. (3.30, 3.31), we arrive at:

$$\delta \mathbf{c}_{\gamma_j} = -\delta \boldsymbol{\mu}_\gamma = -\mathbf{J}_{\mathbf{k}_\gamma} \delta \mathbf{k}_\gamma \tag{C.3}$$

C.2 Nonlinear least-squared estimation to capture $\widehat{\mathbf{k}}_T$

Algorithm 5 Nonlinear LS Estimate

Require: $\mathscr{D}\{(\mathbf{p}_j, \Psi_j)\}_{,\ j=1,\ldots,N}$; $\mathbf{k}_{T_0}$, $(\beta, \eta) > 0$

1: **START** Initialize: $\mathbf{k}_i \leftarrow \mathbf{k}_{\kappa_0}$, $M_{i-1} \leftarrow 1$, $M_i \leftarrow 1 + a\beta, a > 1$

2: **while** $\frac{\|M_{p_i} - M_{p_{i-1}}\|}{M_{p_{i-1}}} \geq \beta$ **do**

3: $M_{p_{i-1}} \leftarrow M_{p_i}$, $\widetilde{\mathbf{c}}_p = \widetilde{\mathbf{c}}_p(\mathbf{k}_i)$, $M_{p_i} = M_{p_i}(\mathbf{k}_i)$,

4: $\mathbf{J}_{c_T} = \mathbf{J}_{c_T}(\mathbf{k}_i)$

5: Update $\mathbf{k}_{i+1}$:

$$\mathbf{k}_{i+1} = \mathbf{k}_i - \mathbf{H}\left(\eta\,(\mathbf{J}_{c_T})^{+}\,\widetilde{\mathbf{c}}_T\right), \eta \in (0,1] \tag{C.4}$$

$$(\mathbf{J}_{c_T})^{+} = \left((\mathbf{J}_{c_T})^{\mathrm{T}}\,\mathbf{W}\,\mathbf{J}_{c_T}\right)^{-1}(\mathbf{J}_{c_T})^{\mathrm{T}}\,\mathbf{W} \tag{C.5}$$

6: **end while**

7: $\mathbf{k}^* \leftarrow \mathbf{k}_i$

Ensure: $\mathbf{k}^*$

C.3 Shape parameter calibration assuming circular bending

An alternative simplified calibration method is applied on the same experimental data sets as comparison baselines, where the circular assumption of the bending shape is used. The calibration problem is simplified significantly to be a linear least-square problem where the only parameter needs calibrating is the length.

$$\mathbf{p}_{xz} = [p_x, p_z]^{\mathrm{T}} = [1 - \sin\theta_e,\ \cos\theta_e]^{\mathrm{T}}\frac{L}{\pi/2 - \theta_e} \tag{C.6}$$

C.4 Details of Simulation Results of Shape Characterization

In chapter 3.7.1, simulation results of shape characterization is presented. A summary is presented in Table 3.2 while data of each simulation condition forms a group and is plotted in Figure 3.10. The data sample points plotted in Figure 3.10 is detailed in the following four tables that correspond to different groups varying simulation condition. In addition,

in the end of this appendix, different noise levels used in simulations are illustrated in Figure C.1.

Table C.1: Simulation results of group 1

Group1 Trials	Pos. Res.	Ang. Res.	Pos. Tar.	Ang.Tar.
1	0.4429	1.9137	0.1143	0.8480
2	0.5532	2.1715	0.1483	0.6360
3	0.5090	1.5298	0.1089	0.4469
4	0.5028	1.0943	0.1742	0.6073
5	0.4987	1.4095	0.0706	0.2750
Average	0.5013	1.6238	0.1233	0.5626
STD	0.0351	0.3792	0.0354	0.1923

Table C.2: Simulation results of group 2

Group 2 Trials	Pos. Res.	Ang. Res.	Pos. Tar.	Ang.Tar.
1	0.4681	1.2662	0.1563	0.6245
2	0.4256	2.1772	0.1194	0.9855
3	0.5023	0.9053	0.0823	0.2292
4	0.4446	0.9511	0.1212	0.3266
5	0.5492	1.5986	0.2380	0.3380
Average	0.4780	1.3797	0.1434	0.5008
STD	0.0439	0.4702	0.0528	0.2760

Table C.3: Simulation results of group 3

Group 2 Trials	Pos. Res.	Ang. Res.	Pos. Tar.	Ang.Tar.
1	2.0728	5.6494	0.3607	1.1459
2	1.6506	5.2655	0.3746	0.9855
3	2.1880	11.0753	1.1446	4.4748
4	2.1608	6.0218	0.7619	2.5439
5	1.8454	1.7762	0.1860	0.4698
Average	1.9835	5.9576	0.5656	1.9240
STD	0.2055	2.9750	0.3454	1.4486

Table C.4: Simulation results of group 4

Group 3 Trials	Pos. Res.	Ang. Res.	Pos. Tar.	Ang.Tar.
1	2.0110	5.8785	0.4153	1.9710
2	1.7096	7.2078	0.3704	3.2315
3	1.9181	5.1509	0.4463	2.9622
4	2.0620	6.6463	0.8688	2.4694
5	1.8127	3.3002	0.1731	0.4125
Average	1.9027	5.6368	0.4548	2.2093
STD	0.1286	1.3595	0.2278	0.9964

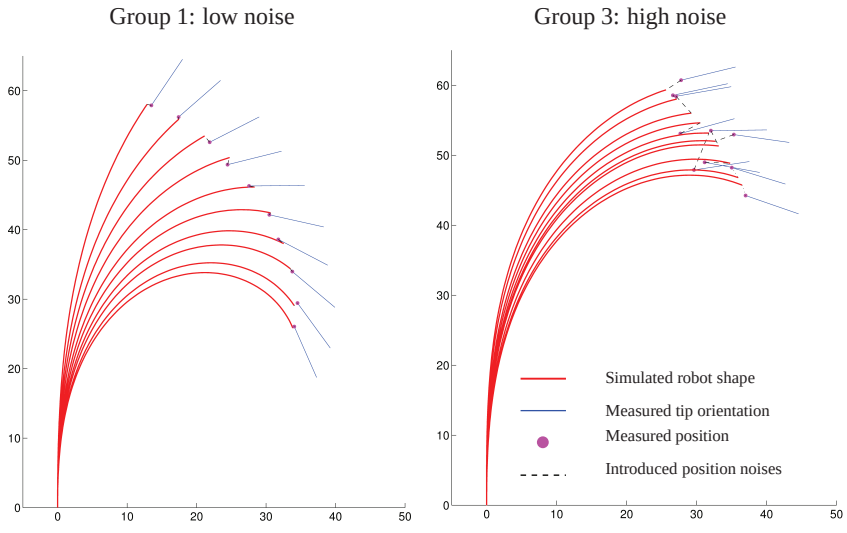


Figure C.1: The pose generation examples from condition 1 and condition 3

Appendix D

Force-Controlled Exploration

D.1 Surface normal estimation using friction estimation

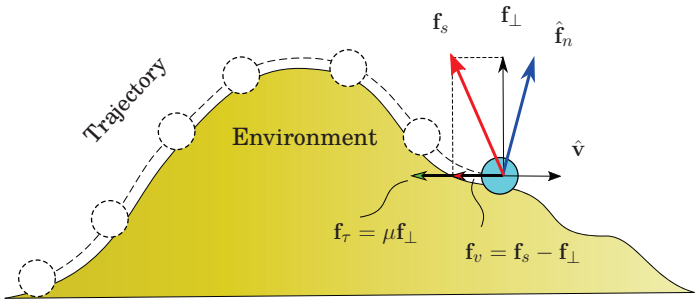


Figure D.1: Surface normal estimation with compensation for environment friction.

This appendix illustrates a potential method to estimate surface normal taking into consideration of the friction effect. The code was implemented in the PSM (Patient Side Manipulator) xPC Simulink model. However the current implementation does not provide satisfactory performance. The online estimated friction keeps overshooting for there is no convergence enforcement. Further investigation including tuning the parameters of the algorithm and adding convergence enforcement may provide significant improvement.

The approach updates the surface normal estimation during the force-controlled exploration and the force controller takes the updated normal direction for control purpose. A simple algorithm is implemented to correct the surface normal force by compensating the environment friction effect.

As illustrated in Fig. D.1 and as detailed in Algorithm 6, a feed-forward coulomb friction $\mathbf{f}_\tau$ is calculated based on a projection force $\mathbf{f}_\perp$ and $\bar{\mu}$, a weighted moving averaged estimation of the friction coefficient. This $\mathbf{f}_\perp$ is the projection of sensed force $\mathbf{f}_s$ onto the null space of robot velocity $\hat{\mathbf{v}}$, and the corrected surface normal force $\hat{\mathbf{f}}_n$ is obtained by subtracting $\mathbf{f}_\tau$ from $\mathbf{f}_s$. The algorithm is executed at 1 KHz, and in each time step (k^{th}), an updated measurement μ_k is calculated using two orthogonal components $\mathbf{f}_\perp$, $\mathbf{f}_v$ and is fed

into the weighted moving average filter. The estimated surface normal direction is obtained by normalizing $\hat{\mathbf{f}}_n$, i.e. $\hat{\mathbf{n}} = \hat{\mathbf{f}}_n / \|\hat{\mathbf{f}}_n\|$.

Algorithm 6 Surface Normal Estimation with Friction Compensation

Given

$\mathbf{f}_s \in \mathbb{R}^3$ ▷ force sensor reading

$\hat{\mathbf{v}} \in \mathbb{R}^3$ ▷ estimated velocity based on previous trajectory

$\hat{\mathbf{M}}_a = [\hat{\mu}_{k-1}, \ldots, \hat{\mu}_{k-m}]^{\mathrm{T}}$ ▷ previous m estimations of friction coefficient

$\mathbf{w} = [w_1, \ldots, w_m]^{\mathrm{T}}$ ▷ moving average weights

Compute $\hat{\mathbf{f}}_n$

if $\|\hat{\mathbf{v}}\| > v_\varepsilon$ **then** ▷ if the robot is moving, v_ε - velocity threshold

  $\bar{\mu} = \frac{1}{m} \sum_1^m w_i \hat{\mu}_{k-i}$ ▷ weighted moving average on $\hat{\mathbf{M}}_a$

  $\mathbf{f}_\perp = (\mathbf{I} - \boldsymbol{\Omega}_v)\mathbf{f}_s,\ \boldsymbol{\Omega}_v = \hat{\mathbf{v}}(\hat{\mathbf{v}}^{\mathrm{T}}\hat{\mathbf{v}})^{-1}\hat{\mathbf{v}}^{\mathrm{T}}$ ▷ projection onto null space of $\hat{\mathbf{v}}$

  $\mathbf{f}_\tau = -\bar{\mu}\,\|\mathbf{f}_\perp\|\,\dfrac{\hat{\mathbf{v}}}{\|\hat{\mathbf{v}}\|}$ ▷ compute friction

  $\hat{\mathbf{f}}_n = \mathbf{f}_s - \mathbf{f}_\tau$ ▷ obtain surface normal force by subtracting friction

else $\hat{\mathbf{f}}_n = \mathbf{f}_s$

end if

Update μ_k

if $\|\hat{\mathbf{v}}\| > v_\varepsilon$ **then**

  $\mathbf{f}_v = \boldsymbol{\Omega}_v \mathbf{f}_s$ ▷ projection onto $\hat{\mathbf{v}}$ direction

  $\mu_k = \dfrac{\|\mathbf{f}_v\|}{\|\mathbf{f}_\perp\|}$ ▷ update current friction coefficient estimation

else $\mu_k = \mu_{k-1}$

end if

Output $\hat{\mathbf{f}}_n$, μ_k

D.2 PSM Force-controlled exploration map

The following figure shows the pre-planned raster scan partitions of the PSM exploration.

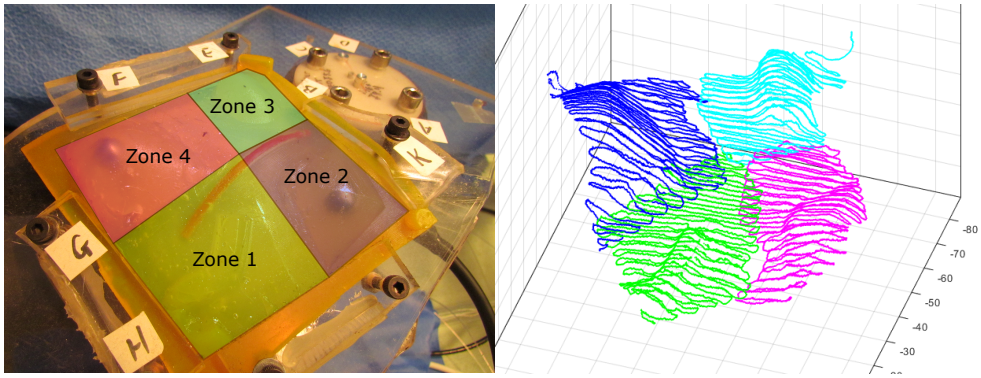


Figure D.2: Raster scan generation partitions of the PSM exploration map

Appendix E

Derivations for Modeling of Continuum Robots with Equilibrium Modulation

E.1 Deriving $\frac{\partial \phi}{\partial \theta}$, $\frac{\partial \phi}{\partial \delta}$, $\frac{\partial \phi}{\partial q_s}$ & $\frac{\partial \phi}{\partial \mathbf{k}_\lambda}$

Rewriting Eq. (4.19) and Eq. (4.16) in a matrix form yields:

$$\underbrace{\begin{bmatrix} k_1+k_2+k_s & -k_1 \\ k_1 & -k_1 \end{bmatrix}}_{\triangleq \mathbf{A}(\psi, q_s, \theta_s)} \begin{bmatrix} \theta_s \\ \theta' \end{bmatrix} = \underbrace{\begin{bmatrix} (k_2+k_s)\,\theta_0 - \lambda \\ k_0(\theta_0-\theta) \end{bmatrix}}_{\triangleq \mathbf{B}(\psi, q_s, \mathbf{k}_\lambda, \theta_s)} \tag{E.1}$$

where $\mathbf{A}$ and $\mathbf{B}$ are defined as functions of $\{\psi, q_s, \theta_s\}$ and $\{\psi, q_s, \mathbf{k}_\lambda, \theta_s\}$ respectively. Using the definition of ϕ, yields:

$$\boldsymbol{A}\left(\underbrace{\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}}_{\triangleq \mathbf{S}_0}\underbrace{\begin{bmatrix} \theta_s \\ \theta_\varepsilon \end{bmatrix}}_{\phi} - \underbrace{\begin{bmatrix} 0 \\ \theta_0 \end{bmatrix}}_{\triangleq \mathbf{C}_0}\right) = \mathbf{B} \tag{E.2}$$

By introducing two constant matrices in the above equation, $\mathbf{S}_0$ and $\mathbf{C}_0$, we have obtained the equation to differentiate:

$$\mathbf{A}\,(\mathbf{S}_0\,\phi - \mathbf{C}_0) = \mathbf{B}, \quad \mathbf{A} \in \mathbb{R}2 \times 2, \mathbf{B} \in \mathbb{R}2, \mathbf{C}_0 \in \mathbb{R}2 \tag{E.3}$$

The full differentiation may be expressed as:

$$(\mathrm{d}\mathbf{A})\;(\mathbf{S}_0\,\phi - \mathbf{C}_0) + (\mathbf{A}\mathbf{S}_0)\,\mathrm{d}\phi = \mathrm{d}\mathbf{B} \tag{E.4}$$

Using $\mathbf{X}'_a$ to denote the partial derivative of matrix $\mathbf{X}$ w.r.t the scalar variable a, i.e. $\mathbf{X}'_a \triangleq \frac{\partial \mathbf{X}}{\partial a}$, then d$\mathbf{A}$ and d$\mathbf{B}$ may be written as:

$$\mathrm{d}\mathbf{A} = \mathbf{A}'_\theta \,\mathrm{d}\theta + \mathbf{A}'_\delta \,\mathrm{d}\delta + \mathbf{A}'_{q_s} \,\mathrm{d}q_s + \mathbf{A}'_{\theta_s} \,\mathrm{d}\theta_s \tag{E.5}$$

$$\mathrm{d}\mathbf{B} = \mathbf{B}'_\theta \,\mathrm{d}\theta + \mathbf{B}'_\delta \,\mathrm{d}\delta + \mathbf{B}'_{q_s} \,\mathrm{d}q_s + \mathbf{B}'_{\theta_s} \,\mathrm{d}\theta_s + \textstyle\sum_i^{n_k} \mathbf{B}'_{k_{\lambda_i}} \mathrm{d}k_{\lambda_i} \tag{E.6}$$

Let us define $\mathbf{C}_\phi$ and $\mathbf{\Gamma}_a$ to provide ease in the derivations:

$$\mathbf{C}_\phi \triangleq \mathbf{S}_0\phi - \mathbf{C}_0, \quad \mathbf{\Gamma}_a = \mathbf{B}'_a - \mathbf{A}'_a\mathbf{C}_\phi \tag{E.7}$$

where the letter $a \in \{\theta, \delta, q_s, \theta_s\}$.

By substituting Eq. (E.5) and Eq. (E.6) into Eq. (E.4), and by using the definitions of $\mathbf{C}_\phi$ and $\mathbf{\Gamma}_a$, we have:

$$\begin{aligned}(\mathbf{A}\mathbf{S}_0) \begin{bmatrix} \mathrm{d}\theta_s \\ \mathrm{d}\theta_\varepsilon \end{bmatrix} - \left[\begin{array}{c:c} \mathbf{\Gamma}_{\theta_s} & \mathbf{0} \end{array}\right] \begin{bmatrix} \mathrm{d}\theta_s \\ \mathrm{d}\theta_\varepsilon \end{bmatrix} = \\ \mathbf{\Gamma}_\theta \,\mathrm{d}\theta + \mathbf{\Gamma}_\delta \,\mathrm{d}\delta + \mathbf{\Gamma}_{q_s} \,\mathrm{d}q_s + \textstyle\sum_i^{n_k} \mathbf{B}'_{k_{\lambda_i}} \mathrm{d}k_{\lambda_i}\end{aligned} \tag{E.8}$$

This equation shows the full differentiation of Eq. (4.35) and all the Jacobians can be obtained directly by their definitions, i.e., the expressions of $\left\{ \frac{\partial \phi}{\partial \theta}, \frac{\partial \phi}{\partial \delta}, \frac{\partial \phi}{\partial q_s}, \frac{\partial \phi}{\partial k_{\lambda_i}} \in \mathbb{R}2 \times 1 \right\}$ may be written as:

$$\frac{\partial \phi}{\partial \theta} = (\mathbf{A}\mathbf{S}_0 - \mathbf{\Gamma}_{\theta_s}\mathbf{S}_1)^{-1} \; \mathbf{\Gamma}_\theta \tag{E.9}$$

$$\frac{\partial \phi}{\partial \delta} = (\mathbf{A}\mathbf{S}_0 - \mathbf{\Gamma}_{\theta_s}\mathbf{S}_1)^{-1} \; \mathbf{\Gamma}_\delta \tag{E.10}$$

$$\frac{\partial \phi}{\partial q_s} = (\mathbf{A}\mathbf{S}_0 - \mathbf{\Gamma}_{\theta_s}\mathbf{S}_1)^{-1} \; \mathbf{\Gamma}_{q_s} \tag{E.11}$$

$$\frac{\partial \phi}{\partial k_{\lambda_i}} = (\mathbf{A}\mathbf{S}_0 - \mathbf{\Gamma}_{\theta_s}\mathbf{S}_1)^{-1} \; \mathbf{B}'_{k_{\lambda_i}} \tag{E.12}$$

where $\mathbf{S}_1 = [1,0]$ is just a selection matrix.

E.2 Derivation of The Jacobian Partitions for the Multi-segment case

Equations (4.54), (4.55) and (4.56) refer to the Jacobian matrix partitions for the two-segment case where the first segment is the inserted subsegment (indicated vby superscript *s*) and the second segment is the empty subsegment (indicated by subscript ε). The Jacobian partitions represent the effects of differentials on θ_i and δ_i that contribute to the end-effector's translational and rotational differential, labeled by subscripts '$\mathbf{v}$' and '$\boldsymbol{\omega}$', indicating 'velocity' and 'angular velocity', respectively. The expressions of $\{\mathbf{J}_{\mathbf{v}\theta_i}, \mathbf{J}_{\boldsymbol{\omega}\theta_i}, \mathbf{J}_{\mathbf{v}\delta_i}, \mathbf{J}_{\boldsymbol{\omega}\delta_i}\}$ are extracted from [103] as:

$$\mathbf{J}_{\mathbf{v}\theta_i} = D_i \begin{bmatrix} c_{\delta_i}\,\chi_{a_i} \\ -s_{\delta_i}\,\chi_{a_i} \\ \chi_{b_i} \end{bmatrix}, \quad \mathbf{J}_{\boldsymbol{\omega}\theta_i} = \begin{bmatrix} -s_{\delta_i} \\ -c_{\delta_i} \\ 0 \end{bmatrix} \tag{E.13}$$

$$\mathbf{J}_{v\delta_i} = D_i \begin{bmatrix} s_{\delta_i}\,\chi_{c_i} \\ c_{\delta_i}\,\chi_{c_i} \\ 0 \end{bmatrix}, \quad \mathbf{J}_{\boldsymbol{\omega}\delta_i} = \begin{bmatrix} c_{\delta_i} s_{\theta_i} \\ -s_{\delta_i} c_{\theta_i} \\ -1 + s_{\theta_i} \end{bmatrix} \tag{E.14}$$

Where $c(\cdot)$ and $s(\cdot)$ denote the cosine and sine functions, and D_i represents the length of the subsegment. For the inserted subsegment, $D_s = q_s$; and for the empty subsegment, $D_r = L - q_s$. In addition, the following shorthanded notations are used:

$$\chi_{a_i} = \frac{(\theta_i - \theta_0)c_{\theta_i} - s_{\theta_i} + 1}{(\theta_i - \theta_0)^2} \tag{E.15}$$

$$\chi_{b_i} = \frac{(\theta_i - \theta_0)c_{\theta_i} + c_{\theta_i}}{(\theta_i - \theta_0)^2} \quad , \quad \chi_{c_i} = \frac{s_{\theta_i} - 1}{\theta_0 - \theta_i} \tag{E.16}$$