THE UNIVERSITY *of* EDINBURGH

Data-driven Modelling and Control of Concentric Tube Robots with Application in Distal Lung Sampling

Balint Thamo

Supervisors: Dr. Mohsen Khadem, Dr. Zhibin (Alex) Li

Doctor of Philosophy

THE UNIVERSITY OF EDINBURGH

2023

Abstract

This research aims to explore the use of Concentric Tube Robots (CTRs) as a novel alternative to needle-based interventions in order to make these procedures more accurate and repeatable. CTRs due to their small footprint, compliance, and dexterity have been proposed for several minimally-invasive robotic surgeries. As a novel flexible robot, it has the potential to reach distal parts of the human lung that are difficult or impossible to reach with conventional needle-based interventions. There are, however, still significant challenges associated with the motion and position control of CTRs. Commonly used model-based control approaches are computationally expensive to solve and often employ simplified geometric/dynamic assumptions, which could be inaccurate in the presence of unmodelled disturbances and external interaction forces. Consequently, this work explores different control strategies to overcome these limitation. This is achieved by first building a simulation environment based on a computationally improved kinematic model that enables real-time control. Then, data-driven control approaches are investigated in order to provide accurate position control in the presence of uncertainties in the system. Finally, a three-phase affordance-aware motion planner is proposed to demonstrate the feasibility of using CTRs for percutaneous lung biopsy.

According to this, the first part of this work concentrates on computationally efficient real-time modelling and simulation of CTRs. In order to achieve this, two approaches are taken. The first one introduces a method that can rapidly estimate the solution of the kinematic model, while the second approach focuses on implementing the existing model in a computationally efficient way in Robot Operating System (ROS) using C++.

Second, this work explores data-driven solutions to control the robot without relying on the kinematic model. Consequently, two data-driven solutions are proposed, namely the Hybrid Dual Jacobian approach and the Extended Dynamic Mode Decomposition (EDMD) algorithm. The hybrid controller combines the advantages of model-based and data-driven control approaches, while the EDMD provides a completely model-free solution to control the robot. Both controllers are capable of rapidly predicting the robot's nonlinear dynamics from a limited data set and offer consistent control under external loading and in the presence of obstacles.

The third part of the thesis explores the use of CTRs in the context of distal lung sampling. This work demonstrates that CTRs are suitable for Needle-Based Optical Endomicroscopy where a CTR steers a fluorescent imaging probe with cellular and bacterial imaging capability inside soft tissue. Then, it is also demonstrated that a CTR can be used as a Steerable Needle to reach a target deep inside the tissue. To achieve these tasks, a motion planner is essential due to the fact that a CTR is only capable of reaching specific points in its workspace and there are a number of configurations where the robot becomes unstable. Based on this, a three-

phase affordance-aware motion planner algorithm is developed. The motion planner selects the best entry point for a specific task. Based on the selected entry point it first generates a stable trajectory from the robot's initial configuration to the selected entry point. Then, a feasible trajectory is generated from the entry point to the target. Finally, the proposed data-driven control algorithm is applied to autonomously steer the robot on the generated trajectory toward the target region for endomicroscopic imaging.

Lay Summary

Lung cancer is one of the most common types of cancer and a leading cause of cancer-related death. Early diagnosis is crucial in order to increase the chances of survival. One of the most common diagnostic methods for detecting early-stage lung cancer is percutaneous lung biopsy. In this method, a needle is guided through the chest wall into a suspicious area to obtain a tissue sample. Unfortunately, this method is not capable of reaching distal areas of the lung and cannot be repeated consistently. The purpose of this thesis is to explore potential alternative methods that could be used to replace this procedure with Concentric Tube Robots. Concentric Tube Robot is a unique type of robot with a miniature size and high dexterity. It is capable of making snake-like movements, which makes it suitable for minimally invasive surgical procedures. The use of Concentric Tube Robots for medical applications has been proposed in a number of studies recently. In spite of the promising results, precise motion control of Concentric Tube Robots is still challenging and it is one of the main limitations of their application. This work explores different ways to control Concentric Tube Robots to enable autonomous steering of the robot for endomicroscopic imaging.

Acknowledgements

First and foremost, I would like to thank my supervisor, Dr. Mohsen Khadem, for his continuous support and guidance throughout my PhD studies. I would then like to express my appreciation to my second supervisor, Dr. Zhibin (Alex) Li and the Surgical and Interventional Robotics Group (SIRG). Furthermore, I would like to thank my sister, Emese, who was always willing to discuss new ideas with me. Finally, I wish to express my gratitude to my family for all the love and support they have provided over the past four years.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

Balint Thamo

Contents

Figures and Tables

Figures

Tables

Chapter 1

Introduction

1.1 Motivation

One of the most common forms of cancer and the leading cause of cancer-related death is lung cancer. It is estimated that the five-year survival rate is around 15%, primarily because it is not detected until an advanced stage has developed. However, early detection of cancer can result in a 70% chance of surviving for 5 years or more [2]. Therefore, the importance of early detection cannot be overstated. The standard method for diagnosing lung cancer is through biopsy, which can either be performed with a needle or with a bronchoscope. A needle-based percutaneous lung biopsy is an invasive procedure performed after a local anaesthetic is given; the doctor uses a needle guided through the chest wall into a suspicious area to obtain a tissue sample. An alternative method is the bronchoscope, a long thin tube equipped with a camera. It is moved down through the main airways of the lungs. Unfortunately, manual bronchoscopy is challenging due to the difficulty of controlling the device inside the inner airways. Thus, the procedure is not repeatable, and its application is limited to the central regions of the lungs, where it is most effective. While both methods rely heavily on the physician's skills, neither is repeatable or capable of reaching distal parts of the lung.

Technological advancements have made it possible to improve the accuracy and repeatability of the above-mentioned procedures by using novel robots. The goal of this research is to investigate the use of Concentric Tube Robots (CTRs) as an alternative method for percutaneous lung biopsy. CTRs or active cannulas are the smallest types of continuum robots (CR). The miniature size and high dexterity of these robots make them attractive for medical applications, such as minimally invasive surgery (MIS). In addition, their snake-like motion makes them ideal candidates for manipulation and navigation within constrained environments, such as those found within the human body.

CRs have been proposed for a variety of minimally invasive procedures such as neurosurgery [3], [4], eye interventions [5], cardiac surgery [6], [7] and lung interventions [8]. A detailed overview of CRs in minimally invasive procedures can be found in [9] and [10].

In spite of the promising results described above, CRs are fundamentally more complex than conventional robots. Their nonlinear dynamics make their modelling and control more challenging, as well as requiring more computational resources. In order to ensure the safe deployment of CTRs during minimally invasive surgeries, precise and reliable motion control is essential.

1.2 Problem Formulation

To achieve CTR-assisted endomicroscopy in percutaneous needle-based interventions the following steps were taken. First, the computational efficiency of calculating the CTR model was improved. Second, data-driven control approaches were examined in order to achieve precise position control. Lastly, a motion planner was proposed to generate feasible trajectories for a given task and to facilitate the safe deployment of CTRs during minimally invasive surgeries.

1.2.1 Computational Improvements to the Kinematic Model

To design control and motion planning strategies for CTRs, it is crucial to calculate their kinematic model. The foundation of the kinematic model was laid down by Dupont et al. [11] and Rucker et al. [12] using the linear constitutive law and the Cosserat rod theory. Based on the proposed kinematic model, the robot's shape can be obtained by solving a set of differential equations with boundary conditions. This results in a Boundary Value Problem (BVP) that can be solved by a shooting method, which is computationally expensive. Two different approaches are taken in order to overcome the high computational cost. First, an observer design is proposed that can rapidly estimate the rod's curvature without explicitly solving the BVP. Second, the kinematic model is implemented in C++ with optimizations and approximations to make it as efficient as possible without compromising accuracy.

1.2.2 Data-driven Control of CTRs

It is often difficult or impossible to achieve precise position control of CTRs with model-based controllers. In the kinematic model of CTRs bending and torsion are dominant while friction, sheer strain and axial elongation are neglected. As a result of the neglected properties and inaccurately identified model parameters, the model's accuracy is limited. Furthermore, the kinematic model does not account for unknown obstacles and external forces. Finally, in order to control the CTR, the robot's inverse kinematics must be calculated. Consequently, a conventional model-based controller for a CTR with 3 tubes will require the model to be calculated 7 times, presenting a challenge for real-time applications. To overcome these difficulties, two data-driven control approaches are proposed.

In the first method, a hybrid control approach is used, in which the model-based Jacobian is used as an initial input for the data-driven controller, and then it is updated according to Broyden's update. This solution combines the advantages of model-based and data-driven approaches. Namely, it has a short learning curve and can adapt to unknown disturbances and uncertainties in the system.

The second method is a purely data-driven solution applying Extended Dynamic Mode Decomposition (EDMD) to approximate the Koopman operator. This holistic approach models the dynamics of a closed-loop controlled CTR. As a result of this method, it is possible to precisely control CTRs in the presence of obstacles without any prior knowledge of the model. Furthermore, its learning curve is very short and can be accomplished before deployment.

1.2.3 Motion Plan for Minimally Invasive Procedures

To demonstrate needle-based interventions with CTRs, it is essential to develop a motion planner that generates a trajectory for specific tasks. CTRs have a limited workspace and there are many configurations where the forward kinematics of the robot loses its uniqueness, which leads to instability. While a precise position controller can provide accurate control locally, a motion planner is necessary to achieve precise control across the robot's workspace. The motion planning goal is to generate a path between the initial configuration and the target position while avoiding unstable configurations and obstacles such as anatomical constraints. Moreover, the tip of the robot is often required to maintain its orientation and to move in the direction it is pointing (local z-direction). Therefore, the robot's workspace is further limited by these requirements. As a result, a three-phase affordance-aware motion planner is proposed, in which the first phase provides all possible entry points for a given task, and the second phase selects the best entry point from the generated entry points and generates a stable path for the task. Finally, the third phase generates a trajectory from the robot's initial configuration to the desired entry point.

1.3 Thesis Statement

In light of the challenges mentioned in the last section, the main objective of this thesis is to explore the use of CTRs in the context of distal lung sampling. Accordingly, this research focuses on answering the following questions:

- How to use novel continuum robots and control algorithms to improve the accuracy of needle-based interventions such as lung biopsy?
- Can the computational cost of the kinematic model be reduced without compromising accuracy?
- Is it possible to achieve precise data-driven control of the robot with a short learning curve in the presence of unknown disturbances?

- Could a purely data-driven approach achieve precise motion control without requiring a long learning phase?

1.3.1 Outline of Contributions

The remainder of the thesis is structured as follows. Chapter 2 provides a literature review. Then, in accordance with the major questions of the thesis, the next four chapters address one of the thesis' central questions. Finally, the results are discussed in Chapter 7.

Chapter 2 presents the relevant background information, including an introduction to continuum robots and CTRs. Following this, position control and motion control strategies are explored for CTRs.

In Chapter 3, the second question of the thesis is examined. Namely, how to improve the computational efficiency of the kinematic model. The kinematic model of CTRs is introduced which is used throughout the thesis. Then two different approaches are taken to improve its computational efficiency. First, it is addressed by a novel nonlinear observer that can rapidly estimate the solution of the Cosserat rod equations. Then, a computationally efficient implementation of the kinematic model is presented, along with a description of the simulation environment and the experimental setup.

In Chapter 4, the major goal is to find a solution to the third question of the thesis, namely how to achieve precise position control in the presence of uncertainties. Accordingly, we discuss how data-driven approaches may overcome these difficulties and the hybrid model-based and data-driven controller is introduced. The proposed controller is then implemented in the simulation environment and the results are experimentally verified.

In line with the main (first) question of the thesis, Chapter 5 examines the use of CTRs for distal lung sampling. A three-phase sampling-based motion planner is presented that can find an optimal entry point (configuration) and provides a stable trajectory for a given task. This is followed by a description of the experimental setup and the verification of the results through experiments. In this chapter, we demonstrate the autonomous steering of CTRs for endomicroscopic imaging.

The final question of the thesis is addressed in Chapter 6 by exploring a purely data-driven strategy with a short learning curve. This controller employs Extended Dynamic Mode Decomposition (EDMD) to learn the nonlinear dynamics of the robot and the interaction forces without using the mathematical model of the robot. In order to verify the controller's performance, simulations and experiments are conducted.

The final chapter of the thesis summarizes the research and explores possible future directions. Data-driven solutions for CTRs are discussed and compared, along with their advantages and disadvantages.

The main chapters of the thesis are based on publications that have been published as part of this research. These publications are listed below:

- Z Mitros, **B Thamo**, C Bergeles, L Da Cruz, K Dhaliwal and M Khadem, "Design and Modelling of a Continuum Robot for Distal Lung Sampling in Mechanically Ventilated Patients in Critical Care", *Frontiers in Robotics and AI* 8, 94.
- **B Thamo**, K Dhaliwal and M Khadem, "Rapid Solution of Cosserat Rod Equations via a Nonlinear Partial Observer", *IEEE International Conference on Robotics and Automation* 2021 (ICRA 2021).
- **B Thamo**, F Alambeig, K Dhaliwal and M Khadem, "A Hybrid Dual Jacobian Approach for Autonomous Control of Concentric Tube Robots in Unknown Constrained Environments", *IEEE International Conference on Intelligent Robots and Systems* 2021 (IROS 2021).
- **B Thamo**, D Hanley, K Dhaliwal and M Khadem, "Data-driven Control of Concentric Tube Robots in Unknown Environments via Dynamic Mode Decomposition", *Robotics and Automation Letters* 2022 (RA-L 2022).
- Emile Mackute, **B Thamo**, K Dhaliwal and M Khadem, "Shape Estimation of Concentric Tube Robots Using Single Point Position Measurement", *IEEE International Conference on Intelligent Robots and Systems* 2022 (IROS 2022).
- **B Thamo**, V Voulgaridou, H Wood, J Stone, K Dhaliwal and M Khadem", Towards Robotics-Assisted Endomicroscopy in Percutaneous Needle-based Interventions", *IEEE Transactions on Medical Robotics and Bionics* 2023 (T-MRB 2023, submitted).

Chapter 2

Background

This chapter provides the necessary background information regarding CTRs. First, an introduction to CRs and CTRs is presented. Then, existing position control strategies and motion planner solutions are examined in the context of CTRs.

The chapter is organized as follows: A brief introduction to CRs is presented in Section 2.1. Section 2.2 provides an overview of CTRs. Then, Section 2.3 discusses control strategies applicable to CTRs including model-based and learning-based solutions. Lastly, existing motion planners are discussed in the context of CTRs in Section 2.4.

2.1 Continuum Robotics

The structure of CRs is fundamentally different from traditional robots. They are a type of high-degree-of-freedom (HDOF) robot inspired by nature. HDOF robots include continuum and hyper-redundant robots. CRs are continuously bending infinite degrees of freedom robots characterized by flexible backbones, while hyper-redundant robots are built from many short rigid links. The research of Hirose laid the foundation of HDOF robots in the 1990s [13]. Based on his work, Chirikjian and Burdick developed robot design and motion control [14], [15] for HDOF robots. Later Gravagne and Walker [16], [17], Hannah and Walker [18] made considerable contributions by creating novel models of continuum and hyper-redundant robots.

CRs are generally classified by the structures of their backbones and their actuation mechanisms. In terms of backbone structure, there are two kinds of continuum robots: single- and multi-backboned. A single-backboned CR is consisted of one elastic element that supports the robot's movement and shape. Meanwhile, multi-backboned CRs have multiple parallel elastic elements at fixed distances from one another. Simaan et al. provide a comprehensive description of these robots in their work [19]. In terms of actuation type, a CR can either be intrinsically or extrinsically actuated. In intrinsic actuation, the actuation is performed within the manipulator itself. Extrinsic actuation occurs outside the manipulator, where mechanical forces are transmitted by the structure. This project focuses on CTRs, which are a special type of CRs. CTRs are single-backboned and manipulated extrinsically. A description of CTRs is presented in the next section.

2.2 Concentric Tube Robots

CTRs or active cannulas are the smallest type of CRs, consisting of at least two concentrically arranged elastic tubes, usually made from Nitinol. Each tube has a straight and pre-curved part. The tubes are nested in each other, and the outer tubes always have larger stiffness than the inner ones. Figure 2.1 shows a CTR with two tubes, which was used in the experiments throughout the thesis. Each tube can make translational and rotational movements, which enables complex motion in constrained environments. A detailed overview of the kinematic model of CTR based on the Cosserat rod theory is given in Section 3.1

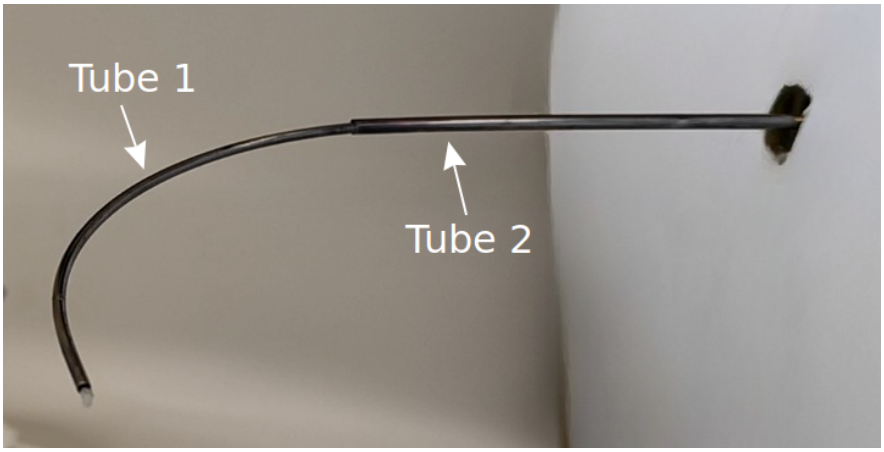


Figure 2.1: An image of a CTR with two tubes used in the experiments. The first tube of the CTR has a 1.1 mm diameter while the second tube has 1.8 mm diameter on the image.

The curvature of the tubes must be carefully considered during the design process. While the robot's motion becomes more complex as the curvature increases, it also becomes more unstable, which leads to mechanical instability, the so-called "snapping" phenomenon. In these cases, elastic potential energy is rapidly released due to the twisting and bending of the tubes. Section 5.2.3 provides further details on CTR instabilities.

The miniature size and high dexterity of CTRs make them attractive for medical applications such as minimally invasive surgeries. There are many kinds of ongoing research for manipulating CTRs within the human body. Cardiac surgery [6], [7], neurosurgery [3], [4], vascular surgery [20], [21], abdominal interventions [22], urology [23], [24] and otolaryngology [25], [26] are just some examples. More applications are available in the survey of Burgner-Kahrs, Rucker [10] and Mitros et al. [9]. While there has been significant progress related to CTRs in recent years, there are still many challenges to be solved to exploit their full potential. Motion and position control are still an open area of research as well as solving the stability problem for CTRs with more than two tubes, especially in the presence of external forces and uncertainties in the model and the environment. It is particularly true when the robot's tip is in contact with tissue.

Throughout the thesis different control solutions are explored where the goal is to alter the robot's joint inputs in order to follow desired trajectories or target positions with the tip of the CTR. The next section gives an overview of different position control approaches for CTRs.

2.3 Position Control of CTRs

The robot's shape and tip pose can be altered by moving the tubes translationally or rotationally. The kinematic model described in Section 3.1 provides the relationship between the joint inputs and the shape of the CTR. Although CTRs are capable of performing complex movements in restricted environments, they are difficult to control and precise control of the robot's motion is essential for the safe deployment of CTRs. The following sections will examine different approaches to control CTRs, including model-based and learning-based approaches.

2.3.1 Model-based Control

CTRs are most commonly controlled by the application of the kinematic model described in Section 3.1. This model provides an approximation of the shape of the robot as a function of joint inputs, *i.e.*, rotation and translation of the tubes. The model is composed of a set of nonlinear differential equations accompanied with boundary conditions that must be solved to find the shape of the robot.

The most common model-based approach for controlling the motion of the CTRs is based on estimating the Jacobian of the robot. The Jacobian can be estimated numerically [27] or analytically [28] using the aforementioned kinematic model. Several researchers have used the Jacobian for open-loop [11], [29], [30] and closed-loop control [27], [31], [32] to control the CTR's motion. These methods assume that the robot is moving in free space and do not consider the effect of external forces acting on the robot as it happens in a realistic constrained environment. Therefore, these methods can potentially cause damage as the robot traverses the anatomy, or be unable to exercise the forces required for tissue manipulation. Researchers have developed dynamic models of CTRs [33] that can simulate the CTR's nonlinear dynamics and accurately estimate robot motion under external forces.

In addition to the aforementioned limitations, models include several parameters representing the mechanical characteristics of the robot that are often not easy to identify independently. Moreover, the models require accurate knowledge of external forces acting on the robot. Unfortunately, these forces are not always known, especially when the robot operates in an unknown environment or is in contact with a deformable object. Therefore, the majority of model-based approaches for controlling the motion of the CTRs assume that the robot is moving in free space [27], [28], [11].

In [34], a radically different approach was used to estimate contacts using real-time medical images and machine learning where the controller employed this feedback to autonomously navigate the inner walls of the heart during cardiac surgery. In other studies, researchers employed CTRs for teleoperated surgical interventions [35], [36]. During the teleoperation, the CTR is remotely-controlled by a user under visual feedback. The user can compensate for controller errors caused by contacts or external forces. In another work [37], a deep neural network was used to estimate robot contact forces at its tip as a function of the robot's shape. This information can be used to update the robot's model for the accurate control of the robot. However, application of these methods are limited where there is a lack of visual feedback or the robot is obscured by anatomical obstacles.

Another approach for safe deployment of CTRs is based on model-based motion planning with obstacle avoidance [38], [39], [40]. The motion planners employ pre-operative medical images to develop a cloud point representing obstacles [38], [39] or create a 3D map of the robot's task space [40]. Next, this data is used to generate collision-free paths for the CTR. Finally, model-based control approaches are employed to steer the robot on the pre-planned path.

The next section provides an overview of learning-based controllers that learn the complex dynamics of the robot and its interaction with the environment to overcome the aforementioned limitations of model-based controllers.

2.3.2 Learning-based Control

The difficulties with the model-based controllers can be overcome by learning-based controllers that learn the complex dynamics of the robot. Commonly, these methods employ machine learning to learn inverse/forward kinematics or dynamic models [41], [42] or learn a direct control policy for moving the robot using Reinforcement Learning (RL) techniques with/without prior knowledge about geometric models [43], [44], [45]. The major disadvantages of these methods are their requirement for numerous training data. For example, a deep-neural-network based approach was proposed in [46] to learn the inverse kinematics of CTRs and a dataset of 100000 samples had to be used to train the network. Gathering large datasets with a real robot is not always feasible. Additionally, relying only on a simulation dataset for training leads to unsatisfactory results when the model is deployed on the real robot [47]. Moreover, these models that are trained offline on experimental datasets [46] or simulated [45] datasets cannot capture the robot's behaviour in contact with the environment or external forces, as it would require a very large training dataset considering numerous robot configurations with various forces.

Some researchers [48] proposed online learning paradigms for the control of flexible robots. These methods require considerable time for the algorithm to converge and learn before reaching the desired accuracy. Additionally, the performance of the algorithm is very susceptible to the quality of feedback signals, which are commonly obtained using electromagnetic trackers or stereo cameras and have a low signal-to-noise ratio. Therefore, these methods often fail in practice due to the poor quality of the feedback signal or the slow learning phase. As a result, none of these methods have been tested on real CTRs. In spite of this, learning-based methods are a fast-growing field, and there are numerous researches on data-efficient algorithms for learning forward/inverse kinematics of robots or control strategies based on reinforcement learning. However, the review of all these methodologies is out of the scope of this work.

2.4 Motion Control of CTRs

The position control strategies described in the previous section provide a method for controlling the robot locally. In order to control the robot across the workspace and to accomplish specific tasks, the CTR requires a motion planner capable of generating a feasible trajectory between the initial configuration and the target configuration. Accordingly motion planners are designed to prevent instabilities and to reach the desired target at a specific pose.

The most common sampling-based algorithms are based on probabilistic roadmaps (PRM) [49] and rapidly exploring random trees (RRT) [50]. There is a wide range of modified versions of these methods that are commonly used in practice, such as RRT* [51] and RRG [52], which converge towards an optimal solution. RRT* generates a tree of valid configurations to find the shortest path from the start position to the target, while RRG creates any graph, not necessarily a tree.

Bergeles et al. generated a stable path with RRT* by discarding the unstable configurations based on their assumptions [53]. Leibrandt et al. demonstrated a framework for stable path planning using PRM [29], while Torres et al. used rapidly exploring random graph (RRG) combining offline and online planners [54]. An RRT-Shape algorithm was also proposed by Wu et al., which uses the CTR's shape as a constraint for the RRT [55]. In interactions where the anatomy of the patient is known, Kuntz et al. proposed a motion planner that utilizes sampling-based motion planning along with local optimization [56]. Sun et al. published a paper describing multiple independent RRTs (MIRRTs), which generated multiple RRTs and selected the one with the lowest estimated probability of collision [57]. These approaches work well for offline path planning in free space.

Chapter 3

Real-Time Implementation of the CTR Model

This chapter focuses on examining different strategies to increase the computational efficiency of calculating the kinematic model of CTRs. First, the kinematic model of CTRs is introduced. Then, two different approaches are investigated in order to provide real-time implementation of the kinematic model. First, a nonlinear observer is proposed that can rapidly estimate the solution of the Cosserat rod equation in the kinematic model. Second, an efficient implementation of the kinematic model is described in detail. Further, the simulation environment is described, which is based on the computationally efficient implementation of the model using Robot Operating System (ROS) with C++ code. This provides a platform for testing different control strategies on the robot. On top of this, communication with the actuators and sensors is also implemented in this platform. Following this, a description of the experimental setup is provided with model parameter identification. Finally, the proposed solutions for computational improvements are compared.

The chapter is organized as follows: Section 3.1 gives a detailed review of the kinematic model of CTRs, which is heavily used for the implementation of the simulation environment. The kinematic model is based on the Cosserat rod theory, which is first applied to a single rod in Section 3.1.1 and then extended to a CTR with multiple tubes in Section 3.1.2. Then, Section 3.3 introduces the proposed nonlinear observer. In Section 3.4 a description of the simulation environment is given along with computational improvements of the kinematic model. This is followed by a description of the experimental setup in Section 3.5 and model parameter identification in Section 3.5.1. Finally, the performance of the proposed methods is investigated and compared in Section 3.6.

3.1 Review of the CTR model

The most common approach for modelling continuum and soft robots is applying the Cosserat rod theory. Cosserat-based models have been developed and experimentally validated for tendon-driven robots [58], CTRs [59], [60], multi-backbone robots [61], and fluidic actuated robots [62]. According to this, first, the Cosserat model is introduced for a single rod. Then it is extended to a CTR with multiple tubes based on the work of Dupont et al. [11] and Rucker et al. [12].

3.1.1 Cosserat Model for a Single Rod

Here, the Cosserat rod equations from [60] are reviewed for a single rod. The following notation is used throughout the thesis: x, $\boldsymbol{x}$, and $\mathbf{x}$ denote a scalar, a vector, and a matrix, respectively. In the Cosserat theory, a rod is modelled as a deformable curve with a frame attached to every point along its arc length, with the z-axis of the frame remaining tangent to the curve. A schematic of a rod under external forces is shown in Fig. 3.1. The configuration of the rod can be defined using a unique set of 3D centroids, $\boldsymbol{r}(s) : [0,\ell] \times [0,\infty] \rightarrow \mathbb{R}^3 \times [0,\infty]$, and a family of orthogonal transformations, $\mathbf{R}(s) : [0,\ell] \times [0,\infty] \rightarrow SO(3) \times [0,\infty]$. For describing the evolution of the curvature along the rod's arc length, the Frenet-Serret frame is used, which provides a moving reference frame to describe the rod's curvature [30]. The curvature can be defined as the magnitude of the rate of change of the unit tangent vector with respect to the arc length or more intuitively the curvature of the rod describes how much the rod bents at a given arc length. Now, assuming the rod is made of linear elastic isotropic materials without pre-twist, we can derive the constitutive equations for calculating the instantaneous curvature of the rod $\boldsymbol{u}(s)$ and the overall shape of the rod:

$$\boldsymbol{r}^{'}(s) = \mathbf{R}(s)\boldsymbol{e}_3, \tag{3.1a}$$

$$\mathbf{R}^{'}(s) = \mathbf{R}(s)[\mathbf{u}(s)]_{\times}, \tag{3.1b}$$

$$\boldsymbol{u}^{'}(s) = -K^{-1}\left[[\mathbf{u}(s)]_{\times}K\big(\boldsymbol{u}(s)-\boldsymbol{u}^{*}\big) + [\boldsymbol{e}_3]_{\times}\mathbf{R}^{T}(s)\boldsymbol{F}\right] \tag{3.1c}$$

where $\boldsymbol{e}_3 = [0,0,1]^T$ is a unit vector aligned with the z-axis of the global coordinate frame, $\boldsymbol{u}^*$ denotes the pre-curvature of the rod in its initial configuration, $\mathbf{K} = \text{diag}(EI, EI, GJ)$ is the stiffness matrix for bending and twisting, E is the rod's Young's modulus, I is the second moment of inertia, G is the shear modulus, J is the polar moment of inertia, $\boldsymbol{F}$ denotes the external loads and $[.]_{\times}$ operator is the isomorphism between a vector in $\mathbb{R}^3$ and its skew-

symmetric cross product matrix:

$$[\boldsymbol{u}]_\times = \begin{bmatrix} 0 & -u_z & u_y \\ u_z & 0 & -u_x \\ -u_y & u_x & 0 \end{bmatrix} \tag{3.2}$$

The boundary conditions for (3.1) are specified in terms of rod's initial curvature $\boldsymbol{u}^*$, rotation α, and translation β of the rod's base.

$$\boldsymbol{r}(0) = [0\ 0\ 0]^T, \tag{3.3a}$$

$$\mathbf{R}(0) = \mathbf{R}_z(\alpha) \tag{3.3b}$$

$$\boldsymbol{u}(\ell+\beta) = \boldsymbol{u}^*, \tag{3.3c}$$

where $\mathbf{R}_z$ denotes a rotation around the z axis and ℓ is the rod's initial length. The boundary conditions given in (3.3) define $\boldsymbol{r}$ and $\mathbf{R}$ at the base of the rod, and curvatures $\boldsymbol{u}$ at the end of the rod, thus forming a boundary value problem. Of note, the rod's curvature at the tip of the rod ($\boldsymbol{u}(\ell+\beta)$) is known from the mechanical parameters of the rod and it is assumed to remain constant.

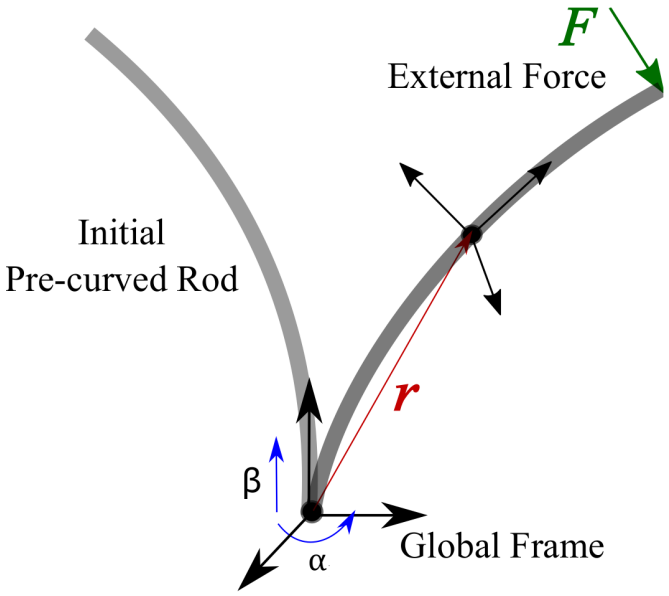


Figure 3.1: A schematic of a Cosserat rod with and without external force. The actuation variables α and β denote the rotation, and translation of the rod's base, respectively.

The model given in (3.1) is quasi-static and the equations are solved in the spatial domain (with respect to s). Then, shooting methods can be used to solve the boundary value problem. A shooting method consists of using a nonlinear root-finding algorithm to iteratively converge on values for $\boldsymbol{u}(0)$, in order to satisfy (3.3c).

3.1.2 Cosserat Model for a CTR

This section explains how the Cosserat model discussed in the previous section can be extended to a CTR with multiple tubes. CTRs are comprised of two or more concentrically arranged pre-curved elastic tubes. Fig. 3.2 shows the schematic of a CTR with three tubes.

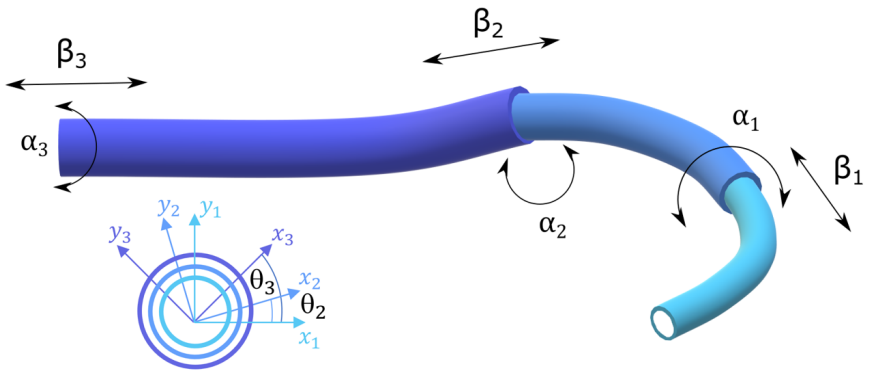


Figure 3.2: Illustration of a concentric tube robot. The robot consists of three concentrically arranged pre-curved elastic tubes. By rotating (α_i) and translating (β_i) the tubes and the shape of the robot can be controlled. θ_i is shown at the robot's cross-section and it denotes the relative twist angle between the tubes around axis z.

In addition to the previous section, the following notations are used. Subscript $i = 1,2..N$ denotes the i^{th} tube of the robot; N is the number of tubes and the outermost one while 1 is the innermost tube. The outer tubes always have larger stiffness than the inner ones. l_i denotes the length of the i^{th} tube, while arc length s is 0 at the proximal end of the tube and l at the distal end. $\boldsymbol{r}(s) : [0,\ell] \rightarrow \mathbb{R}^3$ represent the shape of the robot along arc length s, while $\mathbf{R}(s) : [0,\ell] \rightarrow \mathrm{SO}(3)$ describes the orientation change and twisting along s. $\boldsymbol{q} = (\alpha_1..\alpha_N\beta_1..\beta_N)$ contains the actuation inputs, α_i is the proximal rotation of the base of the i^{th} tube and β_i is the translation of the i^{th} tube. $\boldsymbol{u}_i(s)^*$ and $\boldsymbol{u}_i(s)$ denote the curvature of the i^{th} tube in initial and deformed state respectively. The shape of the robot can be derived from the position vector and the rotation matrix:

$$\boldsymbol{r}'(s) = \mathbf{R}(s)e_3, \tag{3.4a}$$

$$\mathbf{R}'(s) = \mathbf{R}(s)[\mathbf{u}(s)]_\times, \tag{3.4b}$$

where $\boldsymbol{r}'(s)$ is the tangent vector to the curve at arc length s. Each tube conforms to the equilibrium shape of the combined tubes but they are free to twist independently, therefore, the following statements are true:

$$\mathbf{r}_1(s) = \mathbf{r}_2(s) = .. = \mathbf{r}_N(s) \tag{3.5}$$

$$\mathbf{r}'_1(s) = \mathbf{r}'_2(s) = .. = \mathbf{r}'_N(s) \tag{3.6}$$

Following this, segments of the tubes are created. The segments are defined at the end of each tube and at the end of the straight part of each tube. After segmentation, it is possible to estimate the deformed shape of each segment. $\theta_i(s)$ is introduced to parametrize the twist of the tube around axis z, with $\theta_1(s)$ is defined as 0. Therefore, $\mathbf{R}_1(s)$ is identity and

$$\mathbf{R}_i(s) = \mathbf{R}_1(s)\mathbf{R}_{\theta_i}(s) \tag{3.7}$$

where $\mathbf{R}_{\theta_i}(s)$ is the rotation around axis z by $\theta_i(s)$, which is defined by:

$$\mathbf{R}_{\theta_i}(s) = e^{\hat{e}_3\theta_i(s)} \tag{3.8}$$

Applying it to (3.4) the following equation is derived:

$$\mathbf{u}_i(s) = (\mathbf{R}_i^T(s)\mathbf{R}_i'(s))^V = \mathbf{R}_{\theta_i}^T(s)\mathbf{u}_1(s) + \theta_i'(s)\mathbf{e}_3 \tag{3.9}$$

where $\theta_i'(s) = \boldsymbol{u}_{iz}(s) - \boldsymbol{u}_{1z}(s)$ and the operator ^{V} is used to transform an element from $\mathrm{SO}(3)$ into $\mathbb{R}^3$. Based on [12], the curvature of the i^{th} tube can be obtained using the balance of forces and moments in the following way by assuming that each tube conforms to the equilibrium shape of the combined tubes. Please note that the (s) notation is dropped for simplicity.

$$\begin{aligned}\boldsymbol{u}_i'\big|_{x,y} = &-\left(\sum_{i=1}^{N}\mathbf{K}_i\right)^{-1}\sum_{i=1}^{N}\mathbf{R}_{\theta_i}\left[K_i\left(\boldsymbol{u}_i^{'} - \boldsymbol{u}_i^{*'}\right) + [\mathbf{u}_i]_\times K_i\left(\boldsymbol{u}_i - \boldsymbol{u}_i^*\right)\right] - \\ &\left(\sum_{i=1}^{N}\mathbf{K}_i\right)^{-1}\left[[e_3]_\times\mathbf{R}^T(\boldsymbol{F} + \int_0^s \boldsymbol{f}(\varepsilon)d\varepsilon)\right]\Big|_{x,y},\end{aligned} \tag{3.10a}$$

$$\mathbf{u}_{iz}' = \frac{E_iI_i}{G_iJ_i}(u_{ix}u_{iy}^* - u_{iy}u_{ix}^*), \tag{3.10b}$$

where $\boldsymbol{f}$ is the external distributed force applied to the robot; $\boldsymbol{F}$ is the external point load on the robot's tip. Of note, it is difficult to estimate the external forces and this work mainly focuses on data-driven control solutions, therefore, the controllers use a simplified version of the model where forces are assumed to be zero. The effects of known forces are only included as a baseline in the simulation environment where the controller's performance is evaluated. Finally, $\mathbf{K}_i$ is the stiffness matrix for bending and twisting and it is defined as:

$$K_i = \begin{bmatrix} E_iI_i & 0 & 0 \\ 0 & E_iI_i & 0 \\ 0 & 0 & G_iJ_i \end{bmatrix} \tag{3.11}$$

Where E_i is Young's modulus, G_i is the shear modulus of the i^{th} tube. I_i is the second moment of inertia and J_i is the polar moment of inertia of the i^{th} tube's cross-section. Solving (3.4), and (3.10) with the following boundary conditions gives the robot's backbone shape and curvature:

$$\boldsymbol{r}(s)|_{s=0} = \boldsymbol{r}(0) = [0\ 0\ 0]^T, \tag{3.12a}$$

$$\mathbf{R}(s)|_{s=0} = \mathbf{R}_{z(\alpha_1 - \beta_1 u_{1z})}, \tag{3.12b}$$

$$\theta_i(s)|_{s=0} = \alpha_i - \beta_i u_{iz}, \tag{3.12c}$$

$$\boldsymbol{u}_1(s)|_{s=l_i+\beta_i} = \boldsymbol{u}_1^*. \tag{3.12d}$$

3.2 Computational Improvements of the Kinematic Model

Throughout the thesis, the kinematic model described in the previous section is employed. This model forms the basis of the simulation environment presented in Section 3.4. Furthermore, it is utilized by the hybrid controller in Chapter 4 and the motion planner in Chapter 5. Solving the Cosserat rod equations (3.1) is computationally expensive, particularly due to mixed boundary values and kinematic constraints. Due to the high computational cost, lowering the cost of calculating the kinematic model makes real-time applications more feasible. Therefore, computational improvements are required to be made to the model to improve its performance in real-time controllers. According to this, two different approaches are taken to improve the computational efficiency of calculating the kinematic model. The first approach proposes a novel nonlinear observer that can rapidly estimate the solution of the Cosserat rod equations. The second approach implements the kinematic model in an efficient way with optimizations and approximations. The next section concentrates on the implementation of the nonlinear observer. The proposed observer has also been published and the publication with its accompanying media can be found in [63]. Later, the proposed observer was extended to a CTR with 2 tubes by our research group in [64].

3.3 Rapid Solution of Cosserat Rod Equations

The Cosserat model estimates the continuum robots' backbone shape as a function of the robot's mechanical characteristics and known external forces. It consists of several differential equations with boundary conditions split between the base and the tip of the rod. Thus, solving the model involves numerically solving a set of boundary value problems (BVPs), which can be computationally expensive. Several studies [65], [66] including our previous

works [67], [68] have demonstrated implementation of fast enough solutions of the Cosserat model for control of continuum robots. However, the computational cost of the model directly affects the performance and stability of these controllers. As a result, less accurate models with low computational cost are still more attractive [69], [70].

Additionally, there are several promising new designs of continuum robots such as parallel concentric tube robots [61] and eccentric pre-curved tube robots [71] that consists of many kinematically coupled Cosserat rods. The computational cost of the Cosserat rod model is a significant obstacle in the deployment of such designs and more efficient numerical methods are needed.

Motivated by the above discussion, we study the design of a novel observer that can rapidly estimate the solution of Cosserat rod equations without the need to solve the BVP. We propose a nonlinear observer in the sense that the proposed observer employs partial measurement of a Cosserat rod's states (*i.e.*, curvatures at the end of the rod) to estimate the solution of the Cosserat equations. Furthermore, the convergence and stability of the observer are studied and it is shown that the observer predictions exponentially converge to the solution of the BVP. Simulations are performed to compare the performance of the observer with commonly used BVP solvers. Our algorithm is available online[1].

3.3.1 Refined Cosserat Rod Equations

Here the Cosserat model from Section 3.1.1 is refined to be time-dependent, calculating the instantaneous curvature of the rod $\boldsymbol{u}(s,t)$ and the overall shape of the rod can be derived with the constitutive equations in the following way:

$$\boldsymbol{r}^{'}(s,t) = \mathbf{R}(s,t)\boldsymbol{e}_3, \tag{3.13a}$$

$$\mathbf{R}^{'}(s,t) = \mathbf{R}(s,t)[\mathbf{u}(s,t)]_{\times}, \tag{3.13b}$$

$$\boldsymbol{u}^{'}(s,t) = -K^{-1}\left[[\mathbf{u}(s,t)]_{\times}K\big(\boldsymbol{u}(s,t)-\boldsymbol{u}^{*}\big) + [\boldsymbol{e}_3]_{\times}\mathbf{R}^{T}(s,t)\boldsymbol{F}(t)\right] \tag{3.13c}$$

For more details see Section 3.1.1. The boundary conditions for (3.13) are specified in the following way:

$$\boldsymbol{r}(0,t) = [0\ 0\ 0]^{T}, \tag{3.14a}$$

$$\mathbf{R}(0,t) = \mathbf{R}_z(\alpha(t)), \tag{3.14b}$$

$$\boldsymbol{u}(\ell+\beta(t),t) = \boldsymbol{u}^{*}, \tag{3.14c}$$

1. https://github.com/SIRGLab/Rapid-Solution-of-Cosserat-Equations.git

The boundary conditions given in (3.14) define r and $\mathbf{R}$ at the base of the rod, and curvatures $\boldsymbol{u}$ at the end of the rod, thus forming a boundary value problem.

The model given in (3.13) is quasi-static. To solve the equations, it is assumed that at a given time, time-dependent variables are constant and the equations are solved in the spatial domain (with respect to s). Shooting methods can be used to solve the boundary value problem. A shooting method consists of using a nonlinear root-finding algorithm to iteratively converge on values for $\boldsymbol{u}(0,t)$, in order to satisfy (3.14c). Next, the time-dependent variables are updated (*i.e.* $\alpha(t)$, $\beta(t)$), and the equations are solved again in the spatial domain. Our main goal in this work is to design an observer that will employ measurement of $\boldsymbol{u}(\ell+\beta(t),t)$ through time to estimate the correct value of $\boldsymbol{u}(0,t)$ and ensure $\boldsymbol{u}(\ell+\beta(t),t) \to \boldsymbol{u}^* \ \forall t > 0$, without the need to solve the BVP iteratively.

3.3.2 Methodology

In the following sections, we design an observer that can rapidly estimate the rod's curvature $\boldsymbol{u}(s,t)$ in (3.13) without explicitly solving the boundary value problem. Our main assumption is that the solution of (3.13) is unique. We note that a Cosserat rod can buckle under external forces and exhibit elastic instabilities. In this case, the solution of the Cosserat rod equations can oscillate between multiple equilibrium points. In practice, stability measures introduced in [72], [73] can be used to avoid the instabilities and ensure the solution of Cosserat equations remains unique.

3.3.3 Generalized Observable Cosserat Model

Here, we transform the Cosserat rod equations into an observable form that simplifies the design of the observer. To realize the effect of the missing initial value (*i.e.*, $\boldsymbol{u}(0,t)$) on the solution of the Cosserat equations, we define two auxiliary variables, namely,

$$\Gamma(s,t) := \frac{\partial \boldsymbol{u}(s,t)}{\partial \boldsymbol{u}(0,t)}, \tag{3.15a}$$

$$\chi(s,t) := \frac{\partial \big(\mathbf{R}^T(s,t)\boldsymbol{F}(t)\big)}{\partial \boldsymbol{u}(0,t)}. \tag{3.15b}$$

Using (3.15a) and the chain rule, the evolution of the Cosserat rod's curvature in time can be estimated

$$\dot{\boldsymbol{u}}(s,t) = \Gamma(s,t)\dot{\boldsymbol{u}}(0,t). \tag{3.16}$$

We now use (3.13) to derive the equations for calculating $\Gamma(s,t)$ and $\chi(s,t)$. $\Gamma(s,t)$ can be computed by taking the partial derivative of 3.13c with respect to $\boldsymbol{u}(0,t)$.

$$\Gamma^{'}(s,t) = \mathbf{K}^{-1}\Big[[\mathbf{K}(\boldsymbol{u}(s,t)-\boldsymbol{u}^{*})]_{\times}\Gamma(s,t) - [\boldsymbol{u}(s,t)]_{\times}\mathbf{K}\Gamma(s,t) - [\boldsymbol{e}_3]_{\times}\chi(s,t)\Big]. \tag{3.17}$$

In deriving (3.17) we used the following identity

$$\frac{\partial([\boldsymbol{a}]_{\times}\boldsymbol{b})}{\partial \boldsymbol{c}} = -[\boldsymbol{b}]_{\times}\frac{\partial \boldsymbol{a}}{\partial \boldsymbol{c}} + [\boldsymbol{a}]_{\times}\frac{\partial \boldsymbol{b}}{\partial \boldsymbol{c}}. \tag{3.18}$$

We can calculate $\chi(s,t)$ in a similar way. First, we take the transpose of (3.13b). Next, we multiply both sides by $\boldsymbol{F}(t)$. Finally, taking the partial derivative of both sides with respect to $\boldsymbol{u}(0,t)$ gives

$$\chi^{'}(s,t) = [\mathbf{R}^{T}(s,t)\boldsymbol{F}(t)]_{\times}\Gamma(s,t) - [\boldsymbol{u}(s,t)]_{\times}\chi(s,t). \tag{3.19}$$

Remark 1. Based on our assumption of the uniqueness of the solution of the Cosserat rod equations, $\Gamma(s,t)$ defined in (3.15a) is a 3×3 matrix with rank of 3 if $\boldsymbol{u} \neq \vec{0}$. Moreover, based on (3.13) the initial values of $\Gamma(s,t)$ and $\chi(s,t)$ at $s=0$ are $\mathbf{I}$ and $\mathbf{0}$, respectively. Also (3.17) and (3.19) are linear with respect to $\Gamma(s,t)$ and $\chi(s,t)$. Therefore, $\Gamma(s,t)$ and $\chi(s,t)$ are bounded for any bounded s. Considering that s is upper bounded by the length of the rod, both $\Gamma(s,t)$ and $\chi(s,t)$ are bounded for all t.

Now, using (3.13), (3.16), (3.17), and (3.19), we transform the Cosserat rod equations into an observable form.

$$\boldsymbol{r}^{'}(s,t) = \hat{\mathbf{R}}(s,t)\boldsymbol{e}_3, \tag{3.20a}$$

$$\mathbf{R}^{'}(s,t) = \hat{\mathbf{R}}(s,t)[\hat{\mathbf{u}}(s,t)]_{\times}, \tag{3.20b}$$

$$\chi^{'}(s,t) = [\mathbf{R}^{T}(s,t)\boldsymbol{F}(t)]_{\times}\Gamma - [\boldsymbol{u}(s,t)]_{\times}\chi, \tag{3.20c}$$

$$\begin{aligned}\Gamma^{'}(s,t) =& \mathbf{K}^{-1}\Big[[\mathbf{K}(\boldsymbol{u}(s,t)-\boldsymbol{u}^{*})]_{\times}\Gamma(s,t) - \\ & [\boldsymbol{u}(s,t)]_{\times}\mathbf{K}\Gamma(s,t) - [\boldsymbol{e}_3]_{\times}\chi(s,t)\Big],\end{aligned} \tag{3.20d}$$

$$\dot{\hat{\boldsymbol{u}}}(s,t) = \Gamma(s,t)\dot{\boldsymbol{u}}(0,t) \tag{3.20e}$$

with the following boundary conditions.

$$\boldsymbol{r}(0,t) = [0\ 0\ 0]^{T}, \tag{3.21a}$$

$$\mathbf{R}(0,t) = \mathbf{R}_{z}(\alpha(t)), \tag{3.21b}$$

$$\Gamma(0,t) = \mathbf{I}, \tag{3.21c}$$

$$\chi(0,t) = \mathbf{0}, \tag{3.21d}$$

where $\hat{\boldsymbol{u}}(s,t)$ denotes the rod's curvature estimated by the observer and $\mathbf{I}$ is the identity matrix.

3.3.4 Observer Design

Based on (3.14c), the error of the observer in estimating the curvature of the rod at its tip is

$$\boldsymbol{\epsilon}(t) = \hat{\boldsymbol{u}}(\ell + \beta(t), t) - \boldsymbol{u}^* \tag{3.22}$$

In the following theorem, we provide the solution for $\boldsymbol{u}(0,t)$ that drives the error in (3.22) to zero and ensures the solution of the observer in (3.20) converges to the solution of the boundary value problem in (3.13).

Theorem 1. With $\boldsymbol{u}(0,t)$ taken as

$$\boldsymbol{u}(0,t) = -\int_0^t \mathbf{R}\Gamma^T(\ell + \beta(t), t)\mathbf{P}(t)\boldsymbol{\epsilon}(t)\mathrm{d}t, \tag{3.23}$$

where $\mathbf{P}(t)$ is the solution of the differential Riccati equation, which can be used to find the optimal control gain of the system [74].

$$\begin{aligned} -\dot{\mathbf{P}}(t) &= -\mathbf{P}(t)\Gamma^T(\ell + \beta(t), t)\mathbf{R}\Gamma(\ell + \beta(t), t)\mathbf{P}(t) + Q, \\ \mathbf{P}(t_f) &= \mathbf{P}_0, \end{aligned} \tag{3.24}$$

and $\mathbf{Q}$, $\mathbf{R}$, $\mathbf{P}_0$ are all symmetric positive definite matrices and t_f is a design choice, (3.22) is exponentially stable and there exist positive constants c, k, and λ such that

$$\|\boldsymbol{\epsilon}(t_0)\| \leq c \Rightarrow \|\boldsymbol{\epsilon}(t)\| \leq k e^{-\lambda(t-t_0)}, \forall t \geq t_0 \geq 0 \tag{3.25}$$

Proof. Solution of the Riccati differential equation, $\mathbf{P}(t)$, is symmetric positive definite if $\Gamma^T(\ell + \beta(t), t)$ is bounded and non-singular [75]. Based on Remark 1 this condition is satisfied. Now, we select the following Lyapunov candidate

$$V(t) = \boldsymbol{\epsilon}^T(t)\mathbf{P}(t)\boldsymbol{\epsilon}(t). \tag{3.26}$$

Taking the time derivative of V we obtain

$$\dot{V}(t) = \dot{\boldsymbol{\epsilon}}^T(t)\mathbf{P}(t)\boldsymbol{\epsilon}(t) + \boldsymbol{\epsilon}^T(t)\dot{\mathbf{P}}(t)\boldsymbol{\epsilon}(t) + \boldsymbol{\epsilon}^T(t)\mathbf{P}(t)\dot{\boldsymbol{\epsilon}}(t). \tag{3.27}$$

Additionally, from (3.22), (3.23), and (3.20e) we can calculate $\dot{\boldsymbol{\epsilon}}$ as

$$\dot{\boldsymbol{\epsilon}}(t) = \Gamma(\ell + \beta(t), t)\mathbf{R}\Gamma^T(\ell + \beta(t), t)\mathbf{P}(t)\boldsymbol{\epsilon}(t). \tag{3.28}$$

Substituting (3.28) in (3.27) and sorting the equations gives

$$\begin{aligned}\dot{V}(t) =&\boldsymbol{\epsilon}^T(t)\Big(\dot{\mathbf{P}}(t)-\mathbf{P}(t)\Gamma^T(\ell+\beta(t),t)\mathbf{R}\Gamma(\ell+\beta(t),t)\mathbf{P}(t)-\\ &\mathbf{P}(t)\Gamma(\ell+\beta(t),t)\mathbf{R}\Gamma(\ell+\beta(t),t)^T\mathbf{P}(t)\Big)\boldsymbol{\epsilon}(t)\end{aligned} \tag{3.29}$$

Replacing $\dot{\mathbf{P}}(t)$ using (3.24)

$$\dot{V}(t) = \boldsymbol{\epsilon}^T(t)\big(-\mathbf{Q}-\mathbf{P}(t)\Gamma(\ell+\beta(t),t)\mathbf{R}\Gamma(\ell+\beta(t),t)^T\mathbf{P}(t)\big)\boldsymbol{\epsilon}(t) \tag{3.30}$$

$-\mathbf{Q}$ in (3.30) is uniformly negative definite in t by definition. $\mathbf{P}(t)$ is symmetric, thus, $-\mathbf{P}(t)\Gamma(\ell+\beta(t),t)\mathbf{R}\Gamma(\ell+\beta(t),t)^T\mathbf{P}(t)$ has a quadratic form and is negative semi-definite uniformly in t. Hence their sum is negative definite uniformly in t. Thus, $\dot{V}$ satisfies the inequality

$$\dot{V}(t) \leq -\eta\|\varepsilon(t)\|, \tag{3.31}$$

where $\eta > 0$. Based on the Lyapunov Theorem and the foregoing inequality, (3.22) is exponentially stable.

3.3.5 Implementation

The observer given in (3.20) is quasi-static, similar to the Cosserat equations in (3.13). However, it can be solved as an initial value problem using the initial values given in (3.21) and (3.23). At a given time t, time-dependent variables are assumed constant and the equations are solved in the spatial domain using standard methods such as the Runge-Kutta or Adams–Bashforth families of algorithms. Next, the time-dependent variables are updated (*i.e.* $\boldsymbol{u}(0,t)$, $\mathbf{P}(t)$, $\alpha(t)$, and $\beta(t)$). To calculate $\mathbf{P}(t)$, the Riccati differential equation in (3.24) should be solved backwards in time from $t+t_f$, where t_f is the desired finite time, which is selected based on simulation results. The result should be stored in memory. Next, the updated time-dependent variables are used to solve the equations in the spatial domain again. This process is shown in Fig. 3.3.

In the next section, we evaluate the performance of the observer in estimating the solution of Cosserat equations.

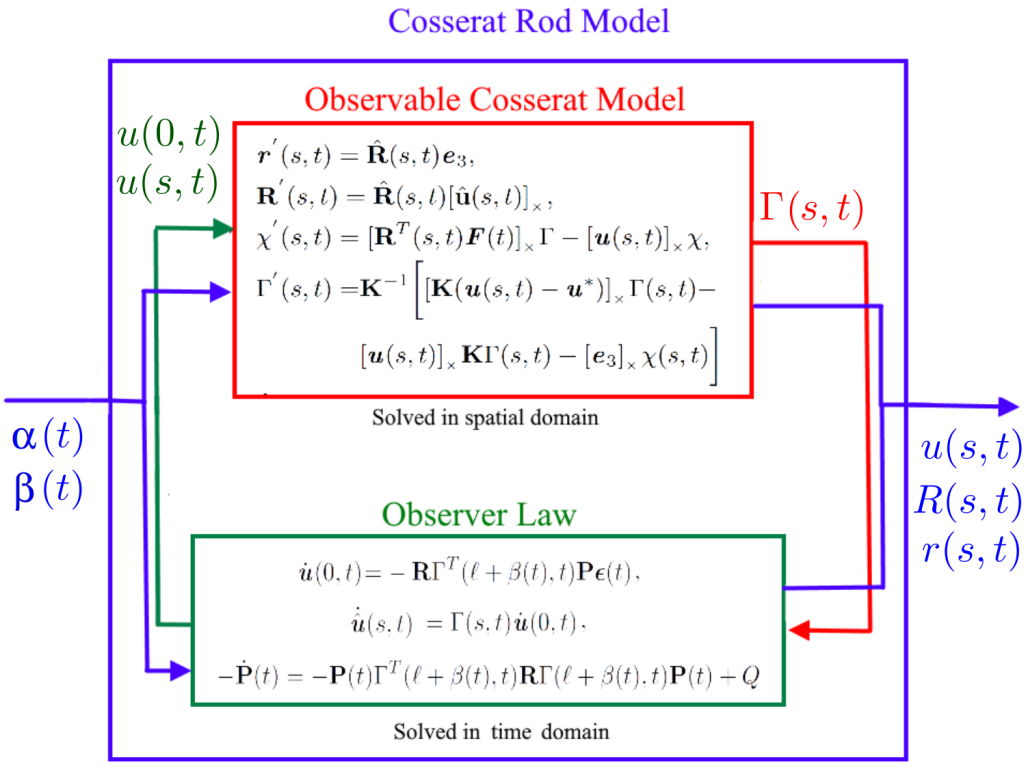


Figure 3.3: A block diagram of the designed observer.

3.3.6 Simulation Study

Simulations are performed to evaluate the proposed observer. Physical parameters of the rod used in the simulations are given in Table 3.1. The parameters are selected from the data sheet of a rod made of Nitinol alloy with outer and inner diameters of 3 mm and 2 mm.

We compared the observer predictions with the solution of the rod equations computed using three different shooting methods. Each method employs a different root-finding algorithm, which to the best of the authors' knowledge, are the most commonly used BVP solvers. These solvers are:

1. Interior point method [76],
2. Quasi-Newton method with BFGS Hessian estimation [77],
3. Nelder-Mead method [78].

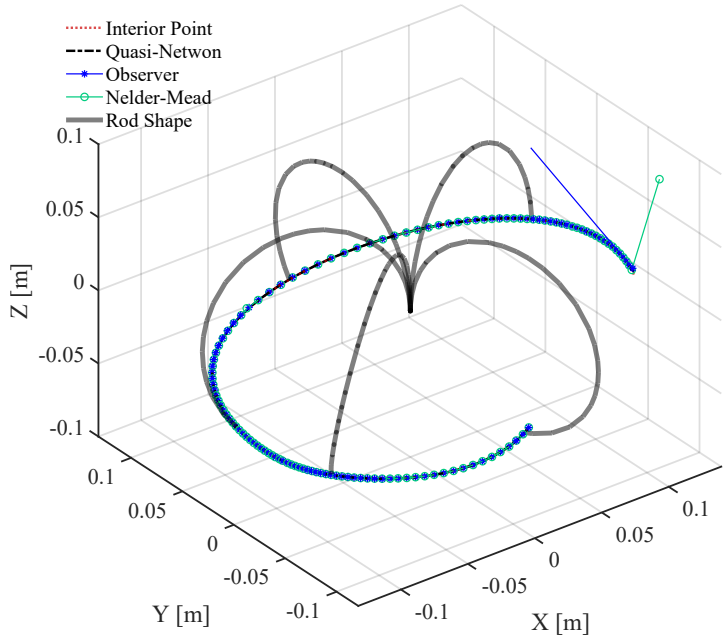
In the simulations, we rotated the rod at a frequency of $2\pi/10$ Hz and pushed the rod at a velocity of 10 mm/sec. Moreover, a time varying force equal to $[\sin(2\pi t/10),\ \cos(2\pi t/10),\ \sin(2\pi t/10)]^T$ was applied to the tip of the rod. The simulation runs for 10 seconds at a sampling frequency of 200 Hz. The observer gains $\mathbf{R}$ and $\mathbf{Q}$ used in the simulations were set to $120 \times \mathbf{I}$ and $30 \times \mathbf{I}$, respectively. These values were found to achieve the minimum prediction error. The tolerance for all the root-finding algorithms was set to 10^{-3}. Moreover, in

all the shooting methods the estimated value of initial curvature at sample time k, *i.e.*, $\boldsymbol{u}(0,t_k)$ was used as the initial guess for the root-finding algorithm in the next step $k+1$. This would make the shooting methods run faster. Moreover, in all the algorithms a 3(2) pair Runge-Kutta formula [79] was used to solve the differential equations governing the motion of the rod. The simulations are performed in Matlab on an Intel Core i7 (2.93 GHz) machine with 16 GB memory.

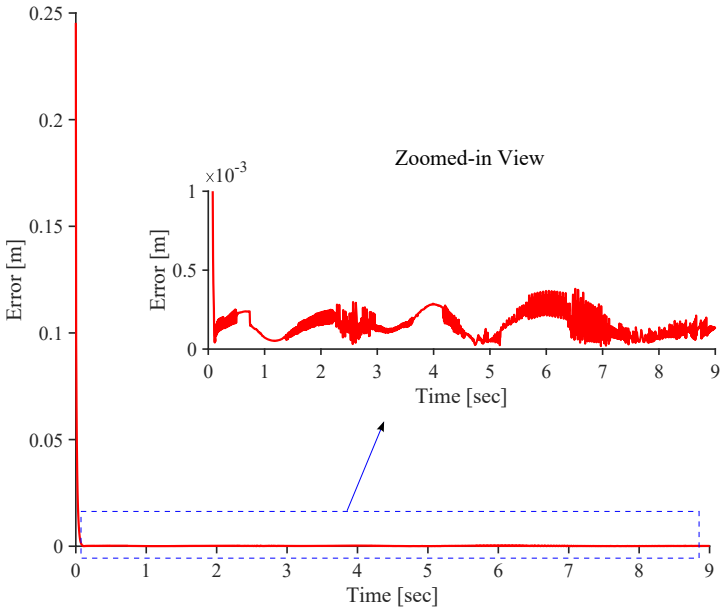
Table 3.1: Physical parameters of the rod. The parameters were selected to have similar properties to CTR tubes described in Section 3.5.1.

ℓ [mm]	400
E [GPa]	$70e9$
G [GPa]	$10e9$
I [m^4]	$3.1907e-12$
J [m^4]	$6.3814e-12$
$\boldsymbol{u}^*$ [m^{-1}]	$[14,5,0]$

Fig. 3.4(a) shows the rod's trajectory estimated via the aforementioned BVP solvers and the observer. As it can be seen, the accuracy of all of the methods is comparable. Also, it can be seen that the observer has an error at the first sampling time but rapidly converges to the correct solution. To investigate the accuracy of the observer, we compared the observer's predictions of the rod's tip position, $\boldsymbol{r}(\ell,t)$ with the results of the interior point method, which was found to be the most accurate BVP solver. The error measured as $\|\boldsymbol{r}(\ell,t)_{\text{observer}} - \boldsymbol{r}(\ell,t)_{\text{interior point}}\|$ is shown in Fig. 3.4(b). It can be seen that the observer rapidly converges to the final solution in less than 0.1 sec. After the convergence, the error of the observer remains below 0.4 mm with an average of 0.13 mm.



(a)



(b)

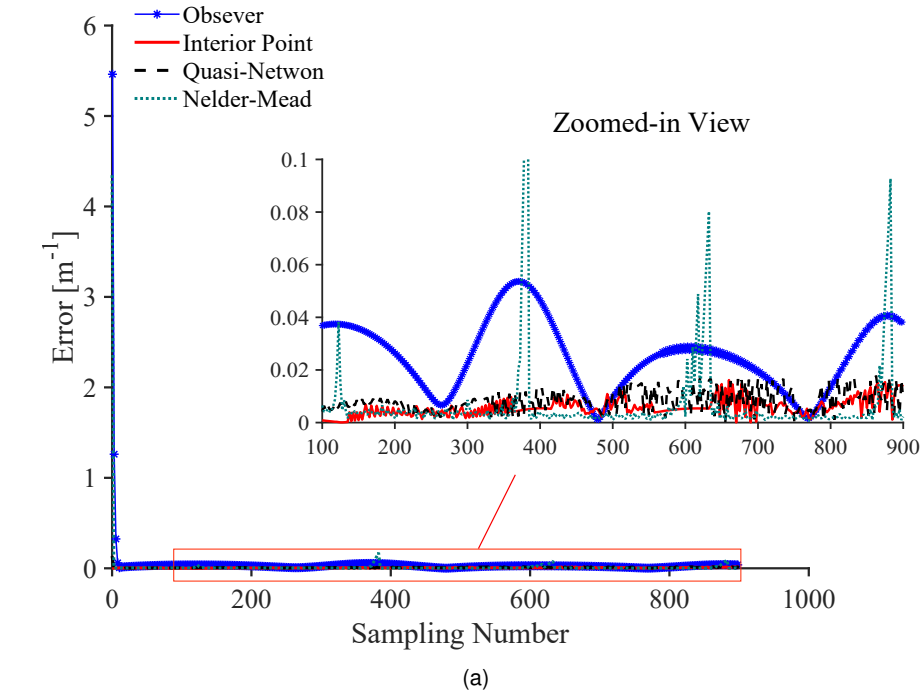
Figure 3.4: (a) A comparison of rod's tip trajectory calculated by solving the rod's model using the observer and 3 different shooting methods. The rod's shape is shown at several configurations along the trajectory. (b) Error of the observer in estimating the position of the rod's tip with respect to the most accurate BVP solver, *i.e.*, interior point method.

Table 3.2: Experimental Results. Mean error (e_{mean}) measured as $\|\boldsymbol{u}(\ell,t)-\boldsymbol{u}^*\|$, the standard deviation of the error (σ_e), the average time to estimate the solution of the model (t_{mean}) for each method, and standard deviation of time σ_t are reported.

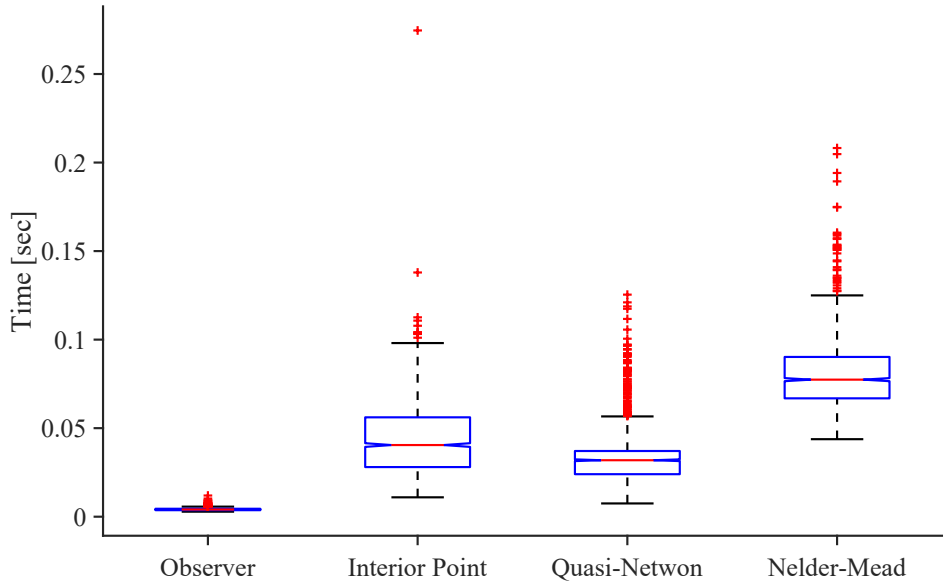
	Observer	Interior-point	Quasi-Newton	Nelder-Mead
e_{mean} [m^{-1}]	0.0241	0.0055	0.0087	0.0101
σ_e [m^{-1}]	0.0537	0.0053	0.0065	0.1034
t_{mean} [sec]	0.0048	0.0525	0.0372	0.0810
σ_t [sec]	$7.4e-4$	0.021	0.0151	0.195

Fig. 3.5(a) shows the error of the solvers and the observer in satisfying the boundary conditions given in (3.14c). The error is measured as $\|\boldsymbol{u}(\ell,t)-\boldsymbol{u}^*\|$. The observer error is in the same order as the BVP solvers and remains below 0.06 m^{-1}. Fig. 3.5(b) compares the computational efficiency of the BVP solvers and the observer in terms of the time that each method takes to compute the solution of the model at each sampling time. The observer is much faster than the BVP solvers and has a lower standard deviation. The average time that the observer takes to estimate the model's solution is 0.0042 seconds, which is significantly faster than other solvers. The fastest BVP method is Quasi-Newton method and can calculate the solution of the Cosserat equations in 0.0330 sec, which is more than 7 times slower.

We performed 20 more simulations, where, rods with different dimensions were moved/rotated at frequencies varying between $\pi/5$ Hz and $\pi/50$. The results are summarized in Table 3.2. The results demonstrate that the observer maintains a similar error as the BVP solvers while exhibiting superior computational efficiency. The mean error of the observer in satisfying the boundary conditions is 0.0241 m^{-1}. Considering the most accurate BVP solver, *i.e*, interior point method, as the ground truth, the observer has a mean error of 0.102 mm in predicting the rod's tip. This error is negligible and much lower than the experimentally validated accuracy of Cosserat equations, which is in the range of 3 to 10 mm [61]. The average time that the observer takes to estimate the model's solution is 7 times faster than the fastest BVP solver, namely, the Quasi-Newton method. The observer can estimate the solution of the Cosserat equations at a sampling frequency of 200 Hz while rendering similar accuracy to slower BVP solvers. The observer can significantly improve the computational efficiency of continuum and soft robots' models that include several kinematically coupled Cosserat rods.



(a)



(b)

Figure 3.5: (a) Accuracy of the observer in satisfying the boundary condition in (3.14c) compared with the BVP solvers. (b) A comparison of the computational efficiency of the observer with common BVP solvers. On each box in (b), the central mark indicates the median, and the bottom and top edges of the box indicate the 25^{th} and 75^{th} percentiles, respectively. The whiskers extend to the most extreme data points and the outliers are plotted individually using the plus symbol.

3.3.7 Concluding Remarks

A new framework for solving the Cosserat rod equations was presented in this section. The Cosserat equations are widely used to model the motion of continuum and flexible robots. However, the computational cost of the model has impeded the widespread application of Cosserat-based models in real-time control of continuum robots. We have demonstrated that our numerical framework can estimate the model's solution 7 times faster than the fastest existing solvers, enabling future applications of Cosserat-based models in real-time control/motion-planning of continuum and soft robots. In the next section, a computationally efficient implementation of the existing model is discussed along with an overview of the simulation environment utilized throughout the thesis.

3.4 Implementation of the Kinematic Model In C++

Based on the kinematic model described in Section 3.1, the CTR model was implemented in Robot Operating System (ROS) using C++. ROS is a popular choice for implementing robotic models. It enables developers to design and test separate nodes for various components of a system. Communication between nodes can be accomplished through messages, where a node can both publish and subscribe to messages. To receive messages broadcast by a publisher node, a receiver node must subscribe to the messages. In addition to this, RViz is used to visualize the simulation results. RViz is a 3D visualization tool developed for ROS.

Based on this the simulation environment is built from the following ROS nodes which is also illustrated in Figure 3.6:

- **ctr_main**: This node is responsible for the communication with the user through the GUI and feeds the **ctr_controller** with the measured tip position, joint positions and requested commands.
- **calculate_model**: The kinematic model of CTR is implemented in this node. Based on the received joint positions and model parameters it calculates and sends back the shape of the robot.
- **ctr_controller**: This is the most important node, where different control strategies are implemented. The node receives the tip and joint position values as well as the mode, action and control parameters from the **ctr_main** node. Based on the selected mode and action it calculates the next movement and sends the requested joint values to the actuators through the **send_motor_commands** node. Finally, it requests the **update_rviz** node to update the robot's shape in RViz.
- **kvaser_interface**: This node provides an interface between a Kvaser USBcan Light 2xHS and ROS. It uses the CANLib library, which enables the communication between ROS and the actuators. The ROS node publishes and subscribes to CAN messages to communicate with the motors' controller.

- **read_tip_position**: The node communicates with the NDI's Aurora System. This system measures the coordinates and orientation of one or more 5DOF electromagnetic sensors with the help of an electromagnetic field generator. After receiving the measured values, it converts them into the CTR's base coordinate system and broadcasted the calculated values to the **ctr_main** node.
- **send_motor_commands**: The purpose of this node is to communicate with the robot's Maxon brushless DC motors (ECXSP16L) through the **kvaser_interface** node. This node broadcasts the read joint positions to the **crt_main** node and forwards the requested joint positions from the **ctr_controller** to the **kvaser_interface**.
- **update_rviz**: RViz is updated from this node. It shows the shape of each tube separately, as well as the overall shape of the robot. In addition to this, the desired tip position and target position markers are visualized here.

To solve the differential equations of the Cosserat model in the **calculate_model** node, we use ODEINT's library, which is included in the Boost C++ libraries [80]. This library provides a wide range of functions to solve ordinary differential equations. In this case, the **integrate_adaptive** function was selected with the Runge-Kutta method to solve the differential equations in the CTR model. In many applications, the tip position of the CTR is sufficient to control the robot. Therefore, it is possible to adjust the precision of the solver in order to reduce the computation time. Once the differential equations have been solved, the robot's shape can be calculated by solving an Initial Value Problem (IVP) or a Boundary Value Problem (BVP). The BVP is solved by using the Levenberg-Marquardt algorithm from the ALGLIB package. Finally, the solution of the BVP gives the shape of the CTR, which is shown in RViz.

The three main goals of the simulation environment are to provide the following things:

- A better understanding of the robot's behaviour.
- A platform for testing different control techniques on the robot.
- A communication channel between the PC and different parts of the experimental setup (motor controllers, limit switches, NDI's Aurora System, electromagnetic sensors).

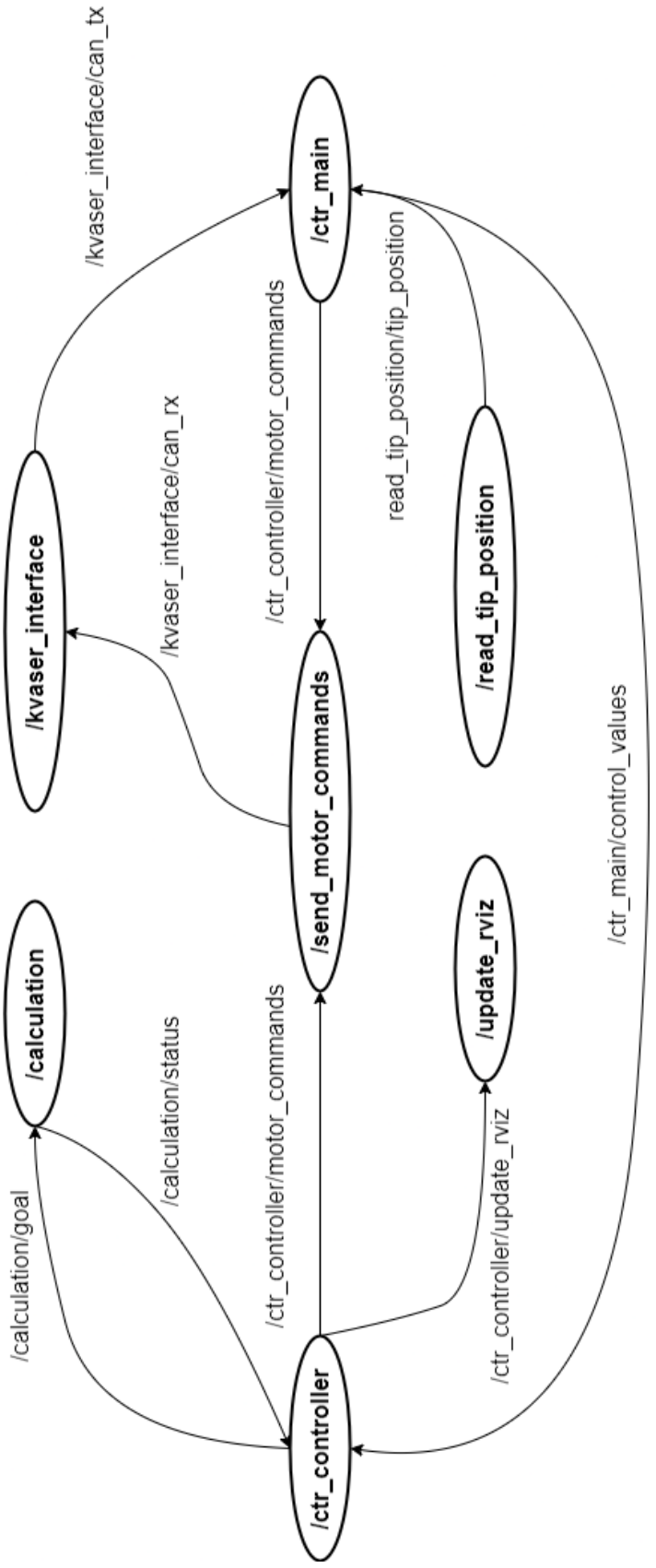


Figure 3.6: ROS nodes and topics in the simulation environment.

As Figure 3.7 illustrates, the user can select the following modes and actions in a simple graphical interface unit:

- Modes
  - Manual
  - Data-Driven
  - Hybrid
  - EDMD
- Controllers
  - PID
  - MPC
- Tasks
  - None
  - Initial Learning
  - Follow Square
  - Follow Half Circle
  - Follow Full Circle
  - Follow Input Trajectory
  - Follow Desired Tip Position

In addition to this, it is also possible to set initial joint inputs (α_i, β_i) and control parameters. Any of the above-mentioned modes and actions can be done on the actual robot and in the simulation environment simultaneously.

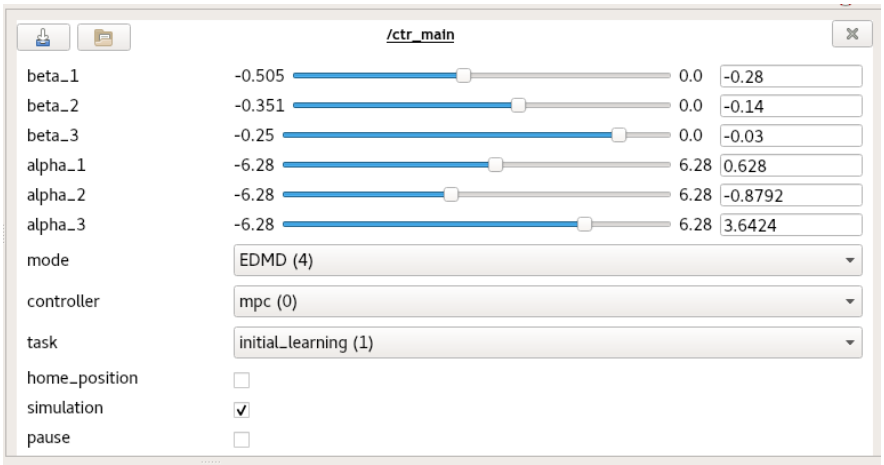


Figure 3.7: The RViz GUI enables the selection of different modes, controllers, and actions as well as setting up initial rotational and translational joint inputs. It is possible to run the implemented controllers on the actual robot and in the simulation environment simultaneously. For example, the initial learning task is selected for EDMD with MPC, which means that an initial learning phase for the EDMD is run to learn the robot's behaviour in a given configuration.

Fig. 3.8 shows a CTR consisting of three tubes using RViz. The shape of each tube is shown separately as well as the overall shape of the robot based on the model. The term robot refers to a CTR in the remainder of this work. The simulation environment also works as a convenient tool for understanding the robot's behaviour. In addition to this, the robot's tip position and target tip position are also shown which helps the user to see how the robot performs and gives insights into the differences between the simulation environment and the actual robot.

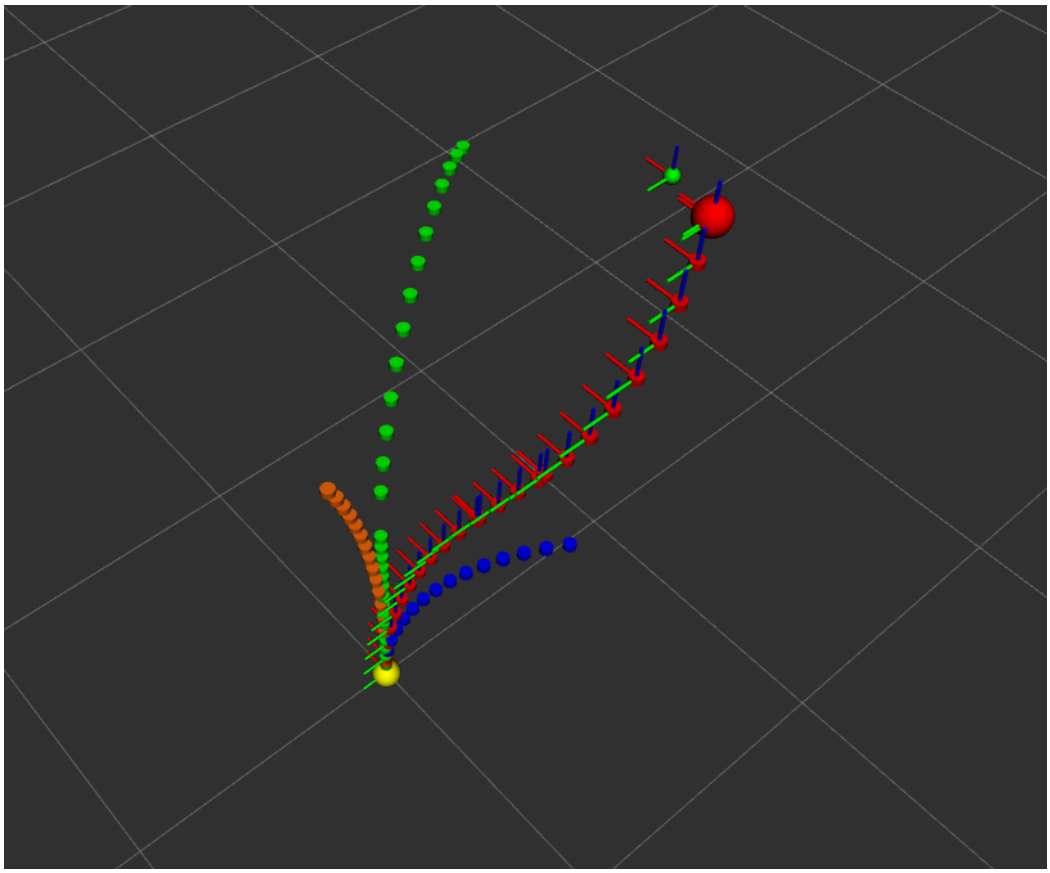


Figure 3.8: RViz shows a CTR with three tubes. It shows the equilibrium shape of the tubes (with local coordinate frames) with the tip position (red sphere), the shape of the first tube (green), the shape of the second tube (blue), the shape of the third tube (orange) and the target position (green sphere).

3.5 Experimental Setup

Our experimental setup is designed based on Section 2.2. The base of the robot is built from 3D printed materials (see Figure 3.9). The robot is actuated by six Maxon brushless DC motors (ECXSP16L) with GPX16HP planetary gearheads and ENX16 encoders. These motors are controlled through six EPOS4 Compact 50/5 CAN digital positioning controllers. Three motors are used for achieving linear motion, and three for rotational movements. Each of the former can move (extend or retract) one tube, while each latter rotates one tube axially. Limit switches are applied for safety reasons to detect linear movement limits and to set reference points to the system. There are no limits for the rotational joints. This setup can be used with two or three tubes. Communication with the motors is done via CAN through a Kvaser USBcan Light 2xHS, making it robust for possible medical applications. It enables communication up to 1 Mbit/s between the PC and the position controllers. The motors used for linear motions are set to the highest possible speed which corresponds to a 1.2 mm/s tip position velocity in Cartesian space. The motors for rotational movements can achieve a much higher velocity than this. Based on this, the desired tip velocity is set to 1 mm/s during the experiments.

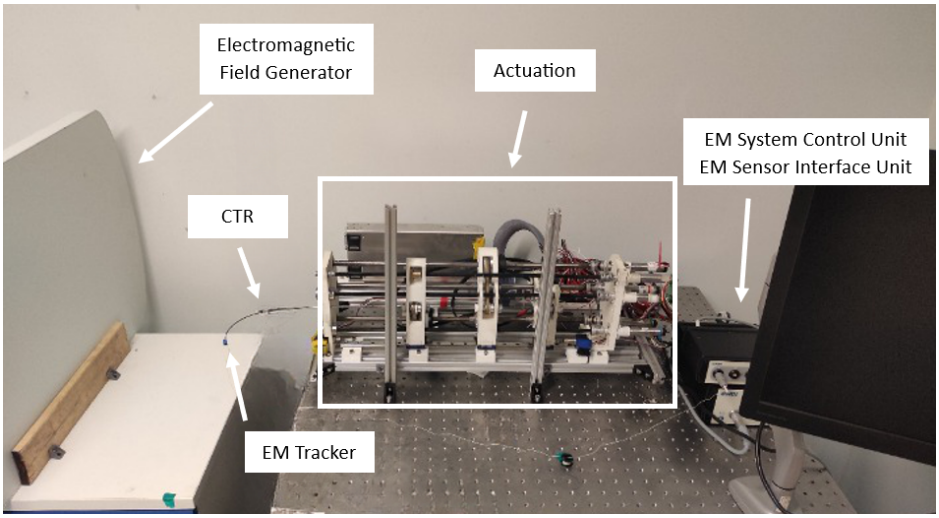


Figure 3.9: Experimental setup for a CTR with 2 tubes. An electromagnetic tracker is placed at the tip of the robot to measure the robot's tip position.

For position and orientation sensing, NDI's Aurora System is used. This electromagnetic tracker consists of a Tabletop Field Generator (FG), a System Control Unit (SCU), a Sensor Interface Unit (SIU) and 5 Degrees of freedom(DoF) electromagnetic sensors. The control unit controls the field generator and receives measurements from the sensor through the sensor interface unit. First, a varying low-intensity electromagnetic field is generated by the field generator. Then, the SIU amplifies the signals received from the sensors and converts them into digital values, which can be read from the PC. For our experiments a single electromagnetic

sensor is used, which is attached to the robot's tip. The sensor has a diameter of 0.55 mm and a length of 8 mm; it is connected with a coil wire to the SIU. The position accuracy of the sensor is 0.7 RMS (mm) with a maximum error of 1.8 mm, while the orientation accuracy is 1.2 RMS ($^\circ$). The Aurora system has a measurement rate of 40 Hz, which is a potential bottleneck of the system.

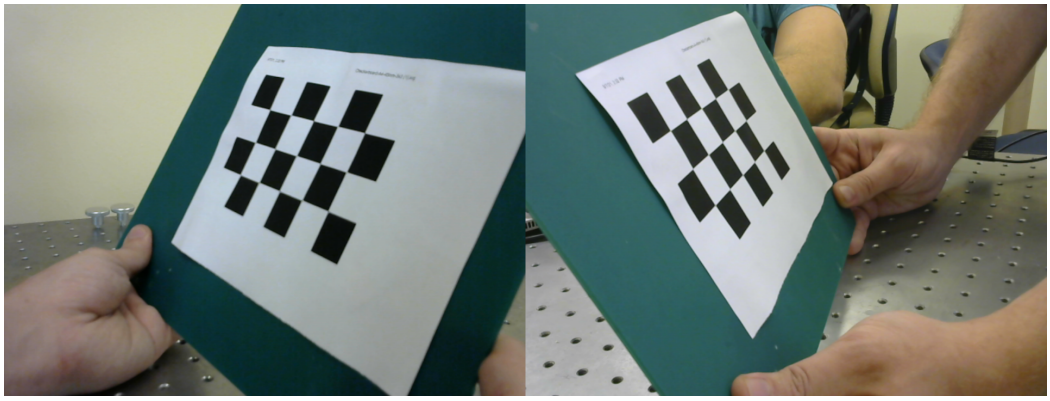
Moreover, the CTR can be equipped with a camera or optical fiber. Enable's minnieScope camera can provide visual feedback to either a software or to the surgeon, while an optical fiber can be used for endomicroscopy.

3.5.1 Identifying Model Parameters

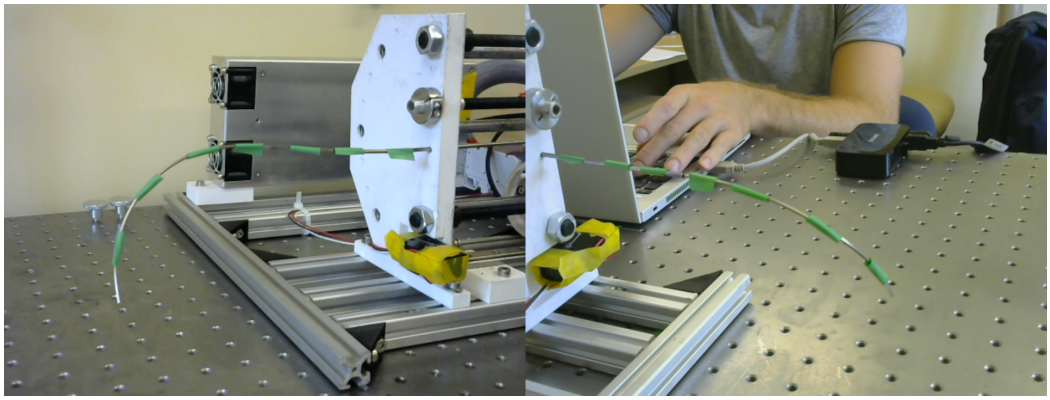
As the kinematic model relies on the model parameters, we first performed a system identification experiment to identify the model parameters. Accurate model parameter identification is important for the following reasons. First, it enables more realistic simulations in the simulation environment. Second, it improves the performance of the hybrid controller in Chapter 4, and improves the quality of the generated motion plans in Chapter 5. Manual backbone segmentation established the base and shape of the CTR relative to the aligned calibration grid. Matching backbone points were selected in both images, and then triangulated to provide a 3-D point cloud (Fig. 3.10). The extracted 3-D backbones were used for calibrating the CTR model parameters, namely, Young's and shear moduli of the tubes. The parameters were identified by fitting the kinematic model given in Section 3.1 to the shape of the robot estimated via the cameras at 25 different configurations. The identified parameters of the model are given in Table 3.3 and are the same as those used in the simulations.

Table 3.3: Mechanical Parameters of the CTR used in simulations.

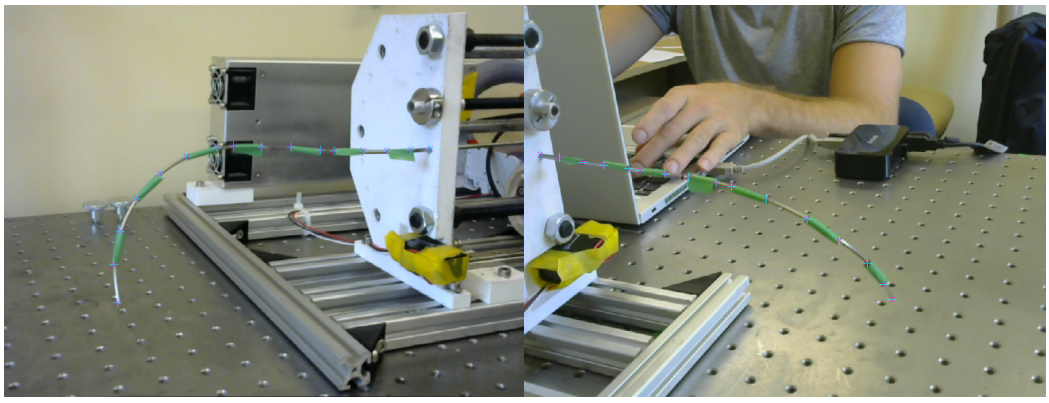
	Tube 1	Tube 2
Inner Diameter[mm]	0.7	1.4
Outer Diameter[mm]	1.1	1.8
Length[mm]	431	332
Curvature[1/m]	21.3	13.1
Young's Modulus, E[GPa]	64.3	52.5
Shear Modulus, G[GPa]	25	21.4



(a)



(b)



(c)

Figure 3.10: Three steps of getting the shape of the CTR using two cameras. (a) Camera calibration using a calibration grid. (b) Taking photos of the robot with two cameras. (c) In both images, matching backbone points are selected and then triangulated to produce a 3-D point cloud.

3.6 Performance of the Implemented Kinematic Model

Various simulation studies were carried out to evaluate the performance of the implemented model. In the integration process, the minimum number of observer calls has a significant impact on the model's performance. The number of observer calls is related to solving the differential equations in the CTR model. It is done by applying the **integrate_adaptive** function, which uses the Runge-Kutta method. The minimum number of observer calls sets the minimum number of required points along each segment when solving the differential equations. A lower minimum number lowers the accuracy of the differential equation's solution but makes the calculation faster. Thus, several simulations were conducted with varying numbers of observer calls in the **integrate_adaptive** function. Furthermore, the computational cost of solving the model as an IVP and a BVP are compared. Of note, a key difference between IVP and BVP is that in IVP the initial conditions are known at the same point (base of the robot), while in BVP, the conditions are split between the base and tip of the robot. Shooting methods are applied to solve the BVP, which require solving the model as an IVP multiple times. Therefore, solving the BVP always takes longer than solving the IVP. The goal of the comparison is to find the optimal way to compute the model in real-time (40 Hz) and to decide whether it is possible to control the robot in real-time, which requires the calculation of the model many times to compute the robot's Jacobian. Of note, the 40 Hz was selected based on the electromagnetic tracker's sampling frequency.

First, a comparison of the accuracy and computational costs of IVP and BVP solvers was conducted for CTRs with two and three tubes. The results of this comparison are summarized in Table 3.4, 3.5 and shown in Figure 3.11. The errors are compared to a baseline provided by the **integrate** function. While the IVP solution provides a fast enough solution for real-time control, its accuracy is much lower than that of the BVP solution. According to the results, the tip position error with the IVP solver is too large for controlling CTRs with three tubes. While the IVP solution enables real-time control of a CTR with two tubes, it is significantly less accurate than the BVP solution.

This was followed by a comparison of the BVP solver for a CTR with two and three tubes as shown in Fig. 3.12. Based on the plots, it can be seen that reducing the number of observer calls between each integration results in a decrease in the computational cost and in the tip position's accuracy. In order to make the real-time calculation feasible, accuracy must be sacrificed. To ensure that the model can be calculated in real-time, the minimum number of observer calls was chosen to be two. In this way, it is possible to compute the kinematic model in real-time at 40 Hz without losing too much accuracy.

In conclusion, the IVP solution provides fast enough computation time for a conventional model-based solver to control a CTR with two tubes in real-time with acceptable accuracy, while the BVP solution allows real-time calculation of the kinematic model with great accuracy. Unfortunately, the high computational cost of this method, however, prevents its use to control CTRs with conventional model-based control strategies in real-time.

Table 3.4: Tip Position Error [m].

Min. number of observer calls	2-Tubes		3-Tubes	
	IVP	BVP	IVP	BVP
1	$1.7e-2$	$1.5e-2$	$4.5e-2$	$2.1e-2$
2	$3.2e-3$	$9.6e-4$	$4.2e-2$	$1.6e-3$
3	$2.4e-3$	$1.8e-4$	$4.2e-2$	$3.4e-4$
4	$2.4e-3$	$5.6e-5$	$4.2e-2$	$1.1e-4$
5	$2.3e-3$	$2.3e-5$	$4.2e-2$	$4.4e-5$
7	$2.3e-3$	$5.8e-5$	$4.2e-2$	$1.1e-5$
10	$2.3e-3$	$1.5e-6$	$4.2e-2$	$2.6e-6$
20	$2.3e-3$	$1.1e-7$	$4.2e-2$	$2.3e-7$
50	$2.3e-3$	$2.2e-9$	$4.2e-2$	$1.1e-8$
100	$2.3e-3$	$2.2e-9$	$4.2e-2$	$1.1e-8$

Table 3.5: Computation Time [ms].

Min. number of observer calls	2-Tubes		3-Tubes	
	IVP	BVP	IVP	BVP
1	1	11	1	17
2	3	19	2	28
3	3	26	2	40
4	4	32	3	52
5	5	39	4	63
7	6	53	5	84
10	7	74	7	120
20	11	138	12	232
50	24	338	26	561
100	44	666	48	1116

3.6.1 Performance Comparison and Discussion

In the previous sections, two different approaches were presented to improve the computational time of calculating the kinematic model of CTRs. The first method introduced a nonlinear observer to estimate the solution of the Cosserat rod equations without solving a boundary value problem. The second method implemented the existing model by applying optimizations and approximations in C++. The development of the simulation environment and the proposed observer happened simultaneously. Several simulation studies were carried out to compare the performance of the two methods. It is important to emphasize that the bottleneck of the experimental setup is measuring the tip pose with the electromagnetic tracker, which has a 40 Hz measurement rate. Therefore, the computational improvement goal is to enable the implemented controller to run at this rate.

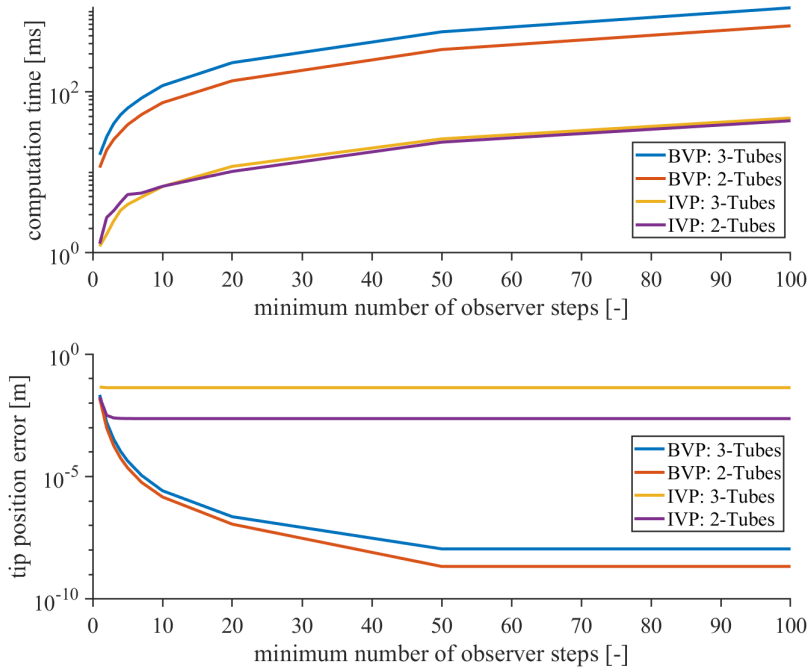


Figure 3.11: Comparison of the implemented BVP solver for 2 tubes and 3 tubes. The plots indicate that the tip position error decreases and the computational time increases with the minimum number of observer calls during the integration.

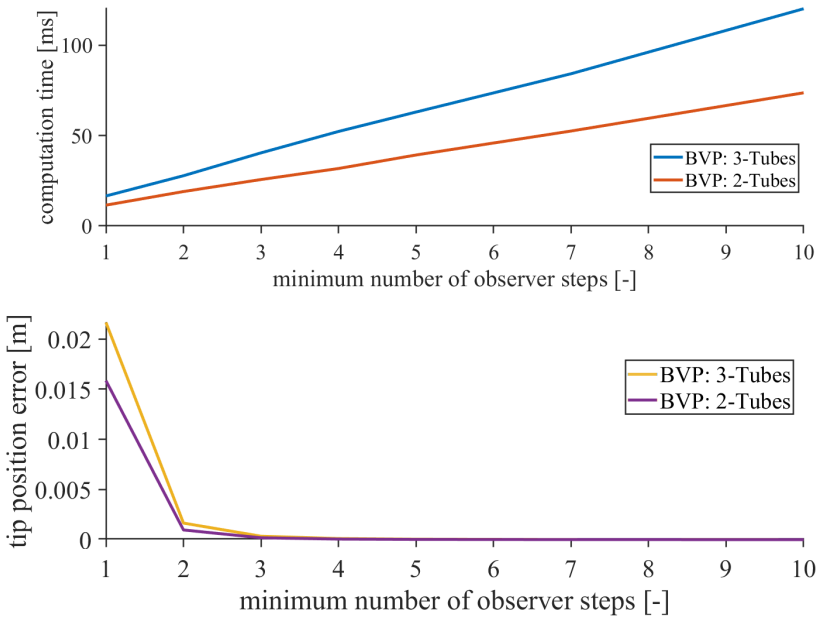


Figure 3.12: Comparison of the implemented BVP solver for 2 tubes and 3 tubes. The plots show that the minimum number of observer calls during the integration affects the tip position error and computational time.

To choose one of the proposed solutions for calculating the kinematic model, the proposed nonlinear observer was compared to the BPV solver for calculating the shape of a rod. An overview of the simulation results is shown in Figure 3.13. Accordingly, the nonlinear observer has a much lower computational cost than the BVP solver, while its tip error is comparable to that of the BVP solution. It should be noted that these simulations were conducted in order to estimate the shape of a single rod rather than a CTR with multiple tubes.

Based on the findings detailed above, the following conclusions were drawn. A 2-tube CTR with the model parameters described in Table 3.3 can be calculated with acceptable error using the IVP solution to achieve real-time control, whereas with the BPV solution, the shape can be accurately determined in real-time. It is not feasible to use IVP or BVP solutions (with conventional model-based controllers) with CTRs that have more than two tubes for real-time applications at 40 Hz using a Desktop Computer with Intel(R) Core(TM) i9-12900K CPU processor and 32.0 GB of Memory. The accuracy of the IVP solution is not acceptable, while the BVP solution is computationally too expensive. The nonlinear observer provided a fast solution with good accuracy in the case of single rods. However, at the time of the decision to choose between the two proposed solutions, the nonlinear observer had only been developed for single rods, and the IVP and BVP solvers had produced satisfactory results for a CTR with 2 tubes. Furthermore, the rest of this work focuses on data-driven control solutions for CTRs that do not heavily rely on the model-based solution. Therefore, we decided to use the

IVP and BVP solutions in the simulation environment. In spite of this, we intend to use the proposed nonlinear observer in our future research for different control approaches and for more complex systems. Accordingly, the proposed observer was subsequently used by our research group to estimate the shape of a CTR with 2 tubes [64].

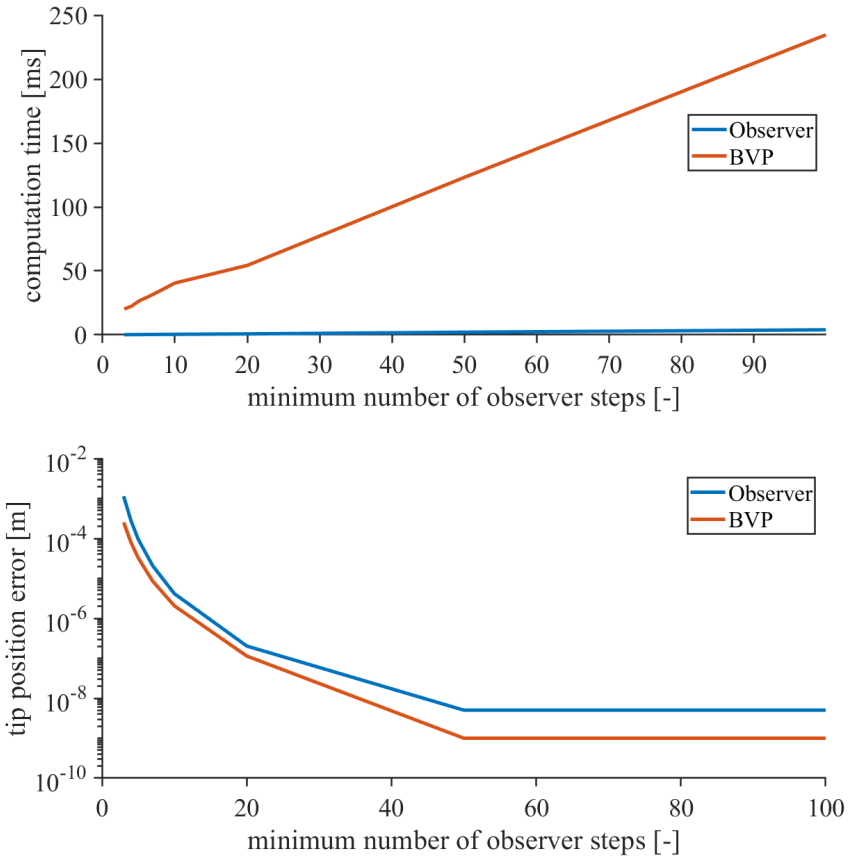


Figure 3.13: Comparison of the implemented nonlinear observer and BVP solver for a single rod. The plots show that both solutions provide satisfactory accuracy but the nonlinear observer is computationally much more efficient.

3.7 Conclusion

The framework of the thesis was laid out in this chapter. Two different approaches were presented to lower the computational cost of calculating the CTR model. First, a nonlinear observer was presented that can rapidly estimate the solution of the Cosserat rod equations without solving a BVP problem. This observer is able to estimate the solution of the model significantly faster than conventional solvers. Then, the simulation environment was implemented in an efficient way in ROS using C++. This simulation environment enables the

users to familiarize themselves with CTRs, run simulations to test different control approaches and communicate with the experimental setup. Following this, the experimental setup was presented with model parameter identification. Finally, a comparison between the nonlinear observer and the computationally efficient kinematic model was carried out. The simulation results show that both solutions provided satisfactory results. Both methods achieved acceptable accuracy, while the nonlinear observer is more efficient than the BVP solution. However, at the time of the decision, the nonlinear observer had only been developed for single rods. As a result, the remainder of the thesis utilizes the computationally efficient implementation of the model. Note, that our research group later extended the use of the nonlinear observer to estimate the shape of CTRs with 2 tubes. The next chapter proposes a data-driven method to control CTRs.

Chapter 4

Hybrid Data-driven Model-guided Control

Providing precise and reliable position control is one of the main challenges of CTRs. It is particularly challenging when there are unknown disturbances and uncertainties in the system. During minimally invasive surgery, constant contact between the robot and the tissue is inevitable. Conventional model-based solvers struggle to adapt to these unknown interactions. In order to overcome this, data-driven control strategies are discussed that can ensure the safe deployment of CTRs. In this chapter, a hybrid controller is proposed, which incorporates Broyden's update with the kinematic model. Following this, a purely data-driven solution based on Extended Dynamic Mode Decomposition is described in Chapter 6.

The main requirement for a data-driven controller is to be able to adapt to unknown external forces and uncertainties in the system while enabling real-time operation at 40 Hz. This sampling frequency was selected based on the electromagnetic tracker's sampling frequency, which is also 40 Hz.

The remainder of the chapter is organized as follows. Section 4.2 describes possible scenarios where data-driven solutions can overcome the limitations of model-based solutions. In Section 4.3 details of the proposed hybrid controller are provided including numerical methods to estimate the model-based and data-driven Jacobians of the robot and the proposed hybrid dual Jacobian control strategy. In Sections 4.4 and 4.5 simulations are performed to verify the performance of the proposed method and results are discussed, respectively. Following this, the hybrid controller is refined in Section 4.6.1 and the simulation results are verified experimentally in Section 4.6.2. Finally, concluding remarks appear in Section 4.7.

The controller proposed in this chapter was also published and can be found with its accompanying media in [81].

4.1 Introduction

To address the challenges of model-based controllers, in this chapter, we propose a hybrid control approach that takes advantage of utilizing a closed-loop data-driven controller initialized by the nominal model of the robot. The proposed method has great potential to overcome the difficulties of the model-based approaches including predicting the effects of unknown external forces, robot dynamics, and unexpected disturbances that might happen in a real surgical setting during the robot's motion. Moreover, to address the challenges associated with current data-driven techniques, it employs the predictions of the kinematic nominal model of the robot to guarantee fast and accurate convergence of the controller. We study the performance of the controller in several scenarios including the robot with inaccurately identified parameters, the robot under external forces, and the robot manipulating a soft-tissue.

4.2 Problem Statement

Commonly used model-based control approaches often employ simplified geometric/dynamic assumptions, which could be very inaccurate in the presence of unmodelled disturbances and external interaction forces. Additionally, the application of emerging data-driven algorithms in real-time control of CTRs is limited due to the fact that these controllers require a considerable amount of time to let the algorithm develop enough to reach the desired accuracy and relevancy.

The goal of this chapter is to present a way to control the motion of a CTR with unknown dynamics in contact with an unknown environment using only measurements of the robot's tip position. A schematic of the CTR is shown in Fig. 4.1, where a CTR is used for the ablation of lung tumours. A CTR carrying an ablation probe is inserted through the chest cavity towards the tumour. The goal is to press the ablation probe on the surface of the tumour without having prior knowledge about the deformation behaviour of the tumour or the tissue surrounding the robot. In such scenarios, generally, there are three sources of disturbance that can lead to inaccurate motion control of the robot:

1. Unknown external forces applied to the robot from the surrounding tissue.
2. Interaction forces between the robot and the soft tissue during tissue manipulation.
3. Inaccuracies in the mathematical model of the robot.

We assume that during the operation, we can track the position of the tip of the robot via commonly used electromagnetic trackers (EMT). It is assumed that the position feedback has a random Gaussian noise. In addition, we assume that the nominal kinematic model of the CTR robot is available, which may cause $\pm 10\%$ error in predicting the position of the robot's tip [27]. We propose an approach to control the CTR's motion in the presence of the

aforementioned disturbances. This novel approach includes three main modules: (i) a model-based estimation of the robot's kinematics in free space, (ii) a data-driven estimation of the robot's motion in the presence of unknown disturbances, and (iii) a hybrid control procedure that fuses these approaches together to accurately and quickly control the robot tip's position in real-time. The following sections describe these modules in detail.

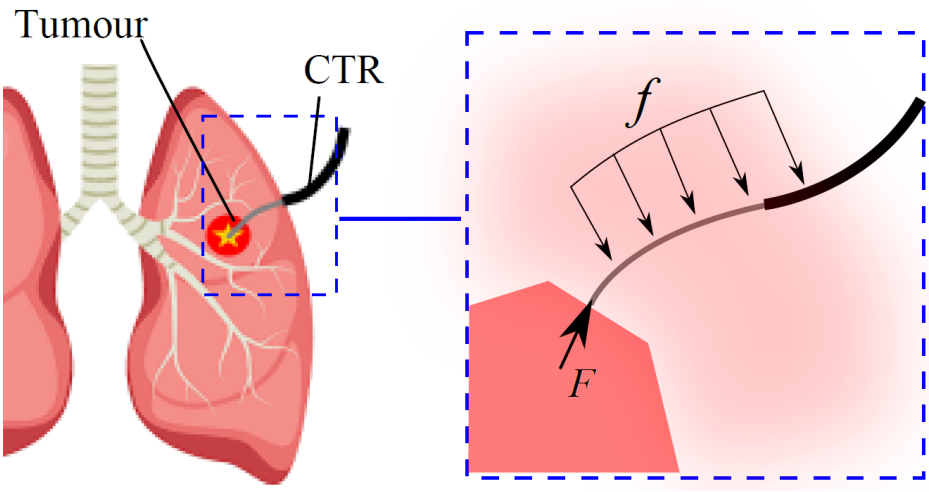


Figure 4.1: A schematic of lung ablation using a concentric tube robot (CTR). Unknown external concentrated point force F and distributed forces f may act on the robot during its motion.

4.3 Methodology

4.3.1 Model-based Jacobian in Free-space

Without loss of generality, we assume that the CTR is composed of three tubes, the final deformed curve of all tubes at a given time t must be equal to the curve of the innermost tube and that the end-effector is the tip of the innermost tube, denoted by $\boldsymbol{x} = \boldsymbol{r}\big|_{s=\ell_1+\beta_1}$, where $\boldsymbol{x}$ is a 3×1 vector denoting the Cartesian coordinates of the robot tip.

For achieving motion control, the Jacobian of the robot is required. It can be numerically estimated using

$$\mathbf{J}_M = \frac{\Delta \boldsymbol{x}}{\Delta \boldsymbol{q}} = \begin{bmatrix} \frac{\boldsymbol{x}^T\left(\boldsymbol{q}+\frac{\Delta q_1}{2}\boldsymbol{e}_1\right)-\boldsymbol{x}^T\left(\boldsymbol{q}-\frac{\Delta q_1}{2}\boldsymbol{e}_1\right)}{\Delta \boldsymbol{q}_1} \\ \vdots \\ \frac{\boldsymbol{x}^T\left(\boldsymbol{q}+\frac{\Delta q_6}{2}\boldsymbol{e}_6\right)-\boldsymbol{x}^T\left(\boldsymbol{q}-\frac{\Delta q_6}{2}\boldsymbol{e}_6\right)}{\Delta \boldsymbol{q}_6} \end{bmatrix}^T, \tag{4.1}$$

where $\boldsymbol{e}_i$ is the i^{th} unit vector of canonical basis of joint space and $\boldsymbol{q} = \left[\beta_1,\ \beta_2,\ \beta_3,\ \alpha_1,\ \alpha_2,\ \alpha_3\right]^T$ is the vector of joint inputs.

We select the above formulation as it gives rise to parallelizable computations without sacrificing in the kinematics model's accuracy. For a three-tubed CTR, the Jacobian is a 3×6 matrix and it maps the joint velocities $\dot{\boldsymbol{q}} \in \mathbb{R}^6$ to the end-effector velocities $\dot{\boldsymbol{x}} \in \mathbb{R}^3$.

The Jacobian given in (4.1), is calculated by numerically linearising the solution of the robot's model given by (3.4) and (3.10). The approximation uses the quasi-static model and the non-linearities in the equations are removed. Moreover, it requires a knowledge of the interaction forces between the robot and the surrounding environment, *i.e*, $\boldsymbol{F} \in \mathbb{R}^3$ and $\boldsymbol{f} \in \mathbb{R}^3$ in (3.10a), which are not always available. Additionally, the Jacobian is susceptible to uncertainties in the identified value of the model parameters, *e.g,* the tubes' stiffness E. It has been shown that the accuracy of the model is equal to 8% of the robot's length due to these uncertainties [12], which means that the predicted shape of the robot has an 8% error of the robot's length compared to the actual shape of the robot. In the next section, we propose a data-driven Jacobian that can overcome the aforementioned difficulties.

4.3.2 Data-driven Jacobian for Free and Constrained Motions

To learn the robot's dynamic behaviour on-the-fly and compensate for the effects of disturbances and external forces, we require an algorithm that updates the elements of the Jacobian in real-time. To this end, we have chosen Broyden's Jacobian update approach that can safely and incrementally learn a CR's Jacobian on-the-fly [82], [83], [84]. This algorithm employs the current measurements of the CTR's tip position $\boldsymbol{x}$, joint inputs $\boldsymbol{q}$, and updates the Jacobian accordingly as follows:

$$\widehat{\mathbf{J}}_D^{k+1} = \widehat{\mathbf{J}}_D^{k} + \chi \frac{\Delta \boldsymbol{x}^k - \widehat{\mathbf{J}}_D^{k} \Delta \boldsymbol{q}^k}{(\Delta \boldsymbol{q}^k)^T (\Delta \boldsymbol{q}^k)} (\Delta \boldsymbol{q}^k)^T, \tag{4.2}$$

where $\widehat{\mathbf{J}}_D^{k+1}$ is the estimated Jacobian matrix at sample time $k+1$, $\Delta \boldsymbol{x}^k$ is the displacement of the robot's tip at sample time k, $\Delta \boldsymbol{q}^k$ represents a vector of actuator input change, and χ is the learning rate ($0 < \chi < 1$).

The advantage of this approach is the ability to adapt to parameter uncertainties and unknown perturbations in the environment due to its online learning behaviour [84], [85], [83]. The main drawback of this method, however, is the long learning time required for estimating the robot Jacobian, particularly when the robot is interacting with an unknown environment [84]. Of note, this is mainly due to the random initialization of the estimated Jacobian matrix and the learning rate parameter χ. As shown in [82], bad initialization of these parameters may cause the algorithm to never converge or need many iterations to estimate a Jacobian matrix.

4.3.3 Hybrid Dual Jacobian

Here, we propose a solution for calculating a novel hybrid Jacobian by combining both the model-based and the data-driven Jacobians (Fig. 4.2). First, the model-based Jacobian is calculated using (4.1). Next, it is used as a weighted initial guess in Broyden's recursive method given in (4.2). The updated equation for estimating the hybrid Jacobian J_H is given by

$$\widehat{\mathbf{J}}_H^{k+1} = e^{-\lambda_1 k}\mathbf{J}_M^k + \frac{1-e^{-\lambda_2 k}}{1+e^{-\lambda_2 k}}\left[\widehat{\mathbf{J}}_H^k + \chi\frac{\Delta\boldsymbol{x}^k - \widehat{\mathbf{J}}_H^k\Delta\boldsymbol{q}^k}{(\Delta\boldsymbol{q}^k)^T(\Delta\boldsymbol{q}^k)}(\Delta\boldsymbol{q}^k)^T\right], \tag{4.3}$$

where $\widehat{J_H}$ and J_M denote the data-driven and the model-based Jacobians, respectively. Also, λ_1 and λ_2 are constant positive scalars working as the weighting factors of the considered Jacobians.

In (4.3), at time zero (*i.e.*, $k = 0$), the data-driven Jacobian $\widehat{\mathbf{J}}_H^k$ is initially set to zero and the model-based Jacobian $\mathbf{J}_M^k$ is used to update the hybrid Jacobian estimation $\mathbf{J}_H^{k+1}$. Then, in the next sample times, the contribution of the model-based Jacobian exponentially decays toward zero at the rate of λ_1, while the weight of the data-driven Jacobian (*i.e.*, $\frac{1-e^{-\lambda_2 k}}{1+e^{-\lambda_2 k}}$) converges toward 1 at an exponential rate of λ_2.

Of note, the proposed hybrid Jacobian is capable of adapting to the unknown perturbations and robot dynamics at a low computational cost, while benefiting from the model-based kinematics to shorten the learning time of the data-driven approach.

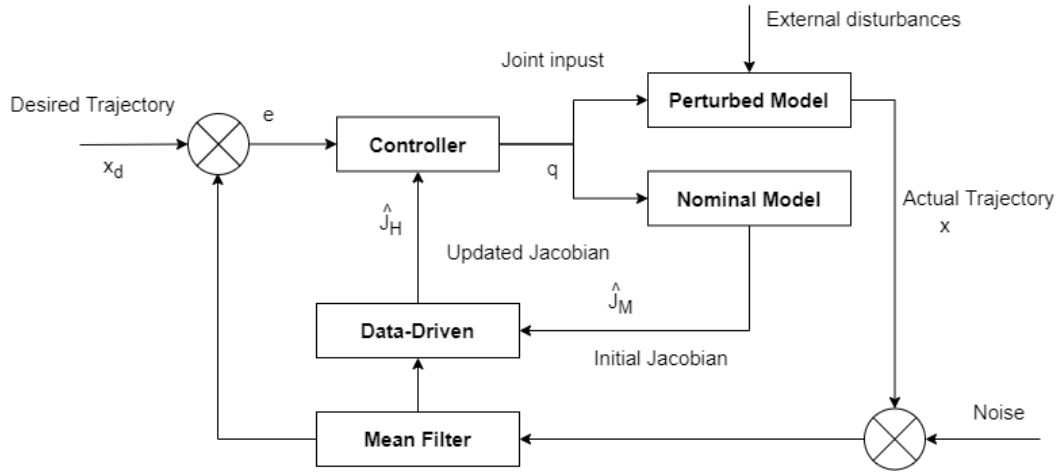


Figure 4.2: Block diagram of the proposed hybrid approach for motion control of CTRs.

4.3.4 Motion Control of CTRs

Using the proposed hybrid Jacobian, we design a controller to steer the tip of the robot to a desired Cartesian position $\boldsymbol{x}_d \in \mathbb{R}^3$. For the CTR robot, this means to find a combination of joint input, *i.e.*, tubes rotation α_i and translation β_i that results in an end-effector position $\boldsymbol{x} = \boldsymbol{x}_d$. To this end, we calculate the forward differential kinematics of the robot as

$$\dot{\boldsymbol{x}} = \mathbf{J}_H(\boldsymbol{q})\dot{\boldsymbol{q}}. \tag{4.4}$$

By introducing the pseudo-inverse of the hybrid Jacobian [86],

$$\widehat{\mathbf{J}}_H^{\dagger} = \mathbf{J}_H^T(\mathbf{J}_H\mathbf{J}_H^T)^{-1}, \tag{4.5}$$

The pseudo-inverse is used since $\mathbf{J}$ is not a square matrix. Equation (4.4) can then be modified to obtain the inverse kinematics of a CTR:

$$\dot{\boldsymbol{q}} = \widehat{\mathbf{J}}_H^{\dagger}\dot{\boldsymbol{x}}. \tag{4.6}$$

To follow a desired trajectory $\boldsymbol{x}_d$, an error between the desired and actual tip position is introduced: $\boldsymbol{e} = \boldsymbol{x}_d - \boldsymbol{x}$. Equation (4.6) can be converted to the following control law to minimize this error:

$$\dot{\boldsymbol{q}}_d = \widehat{\mathbf{J}}_H^{\dagger}\left[\dot{\boldsymbol{x}}_d + \mathbf{K}_P\boldsymbol{e}\right], \tag{4.7}$$

where $\mathbf{K}_P$ is a 3x3 symmetric positive definite matrix containing the controller gains. Fig. 4.2 shows a block diagram of the proposed motion control approach.

4.4 Simulation Studies

To evaluate the performance of the proposed approach, we compared the proposed hybrid dual-Jacobian controller with two commonly used controllers: (i) a model-based controller proposed in [27], and (ii) a data-driven controller employing the Broyden's regression algorithm [83], [84] via three different types of simulations. In the performed simulations, a nominal model of the CTR was first used to calculate the model-based Jacobian in free space via (4.1). Of note, this nominal model was derived based on an existing robot used in [27]. The model parameters of this simulated CTR have been summarized in Chapter 3 (Table 3.3). Next, the model-based Jacobian was used as (4.3) to calculate the hybrid Jacobian during the performed simulations and on-the-fly. Finally, the control law given in (4.7) is employed on the hybrid Jacobian to steer the robot on a pre-defined desired trajectory. The algorithm was tested on a perturbed CTR model in an unknown environment. This perturbed model simulates the effects of unknown disturbances including unknown external distributed/point

loads. To make the simulation more realistic, a normally distributed random noise is added to the robot's tip position:

$$n = \frac{1}{\sigma\sqrt{2\pi}}\exp\left(\frac{-x^2}{2\sigma^2}\right), \tag{4.8}$$

with $\sigma = 2$ mm. Later, a mean filter is used to denoise the feedback signal. The filter takes the average of 5 consecutive samples to reduce the noise.

To evaluate the proposed hybrid approach, the following different simulation scenarios were considered while using three different perturbed models:

(S_1) *Robot with inaccurately identified parameters:* The controller is tested on a perturbed CTR model with $\pm 20\%$ inaccuracy in initial pre-curvature of the robot $\boldsymbol{U}$, tube's Young's Modulus (E), and Shear Modulus (G) of the tubes. These inaccuracies result in a maximum of 10% error in estimating the robot's tip position [27].

(S_2) *Robot under unknown external forces:* The controller was tested on a CTR model that includes external forces. A point load equal to $[0.5, 0.5, 0.5]^T$ N was applied to the tip of the robot. The controller has no prior knowledge of this force.

(S_3) *Robot manipulating a soft tissue:* The CTR robot is in contact with a soft tissue without any *a priori* knowledge of the tissue's mechanical characteristics. This study simulates the tissue ablation procedure, where a probe capable of applying intensive heat/cold is pressed and moved on the surface of the tissue to destroy abnormal tissue. In the performed simulation, the robot is commanded to deform the surface of the tissue by a magnitude of 1 mm and move along a rectangle while keep pushing the tissue by the same amount. To simulate this scenario, the deformation behaviour of the tissue was modelled using the soft tissue model presented in [87]:

$$F = a_1 x + a_2 x^2 \tag{4.9}$$

where F is the contact force between the robot and the tissue and x is the relative surface deformation of the tissue. Also, $a_1 = 0.03$ $\frac{\text{N}}{\text{mm}}$ and $a_2 = 0.003$ $\frac{\text{N}}{\text{mm}^2}$ are constants representing the mechanical characteristics of kidney and liver tissue [87]. In addition to the mentioned contact force F, a time-varying distributed force f was also applied along the whole shape of the robot:

$$f = \sin\left(\frac{2\pi t}{2}\right), \tag{4.10}$$

where t is time. Of note, this distributed force only has a y component with a maximum value of 1 $\frac{\text{N}}{\text{m}}$. The controller has no prior knowledge of this force.

For all three scenarios, the nominal model of the robot was fixed and no external forces were considered during the calculation of the model-based Jacobian. Of note, the external forces were calculated in the perturbed model to give a baseline for the simulation. Also, the robot's tip was steered to follow a square trajectory with 35 mm base length in the first two cases and a 60×30 mm rectangle trajectory in the third case, which simulates the size of a lung tumour at stage 2 or 3 [88]. A square trajectory was selected since the CTRs have more difficulties in traversing straight lines and sharp corners as opposed to circular paths.

Using C++, we simulated these scenarios in Robot Operating System (ROS). Simulation sampling time and the robot's tip desired velocity were selected as 15 millisecond and 5 $\frac{\text{mm}}{\text{sec}}$, respectively. In addition to these, the following controller parameters were chosen: $\mathbf{K}_p = 3\mathbf{I}$, where $\mathbf{I}$ is a 3×3 identity matrix. Also, we selected λ_1, λ_2, and χ in (4.3) as 0.1, 0.1, and 0.1, respectively. These values were tuned via trial and error to achieve the minimum tracking error. Fig. 4.3 shows the robot in free space and in touch with the tissue in the ROS simulation environment. The joint inputs and the corresponding tip positions were used for the data-driven and hybrid controller to learn the behaviour of the robot.

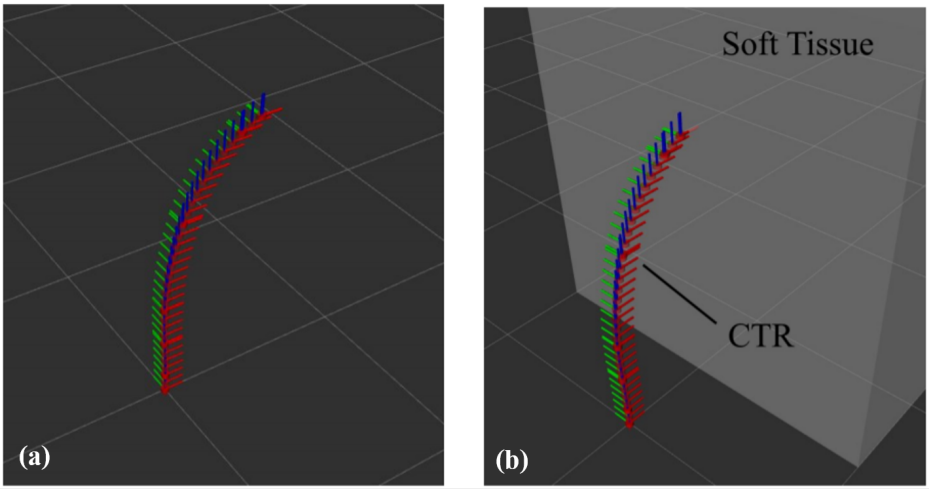


Figure 4.3: CTR simulated in ROS environment. (a) CTR in free space, (b) CTR in contact with soft tissue.

4.5 Results and Discussion

Results of the three scenarios are summarized in Table 4.2. The average tracking error, the standard deviation of the error, the maximum tracking error, and the root-mean-square error (RMSE) are reported. The root-mean-square error is calculated as

$$\text{RMSE} = \sqrt{\frac{\sum_{j=1}^{m} (\boldsymbol{x}_j - \boldsymbol{x}_{d,j})^2}{m}}, \tag{4.11}$$

Table 4.1: Comparison of Error between desired and actual trajectories in three scenarios. Results of model-based, data-driven, and hybrid controllers are presented. e_{mean} is the average error, σ_e is the standard deviation of the error, e_{max} is the maximum error, and RMSE is the root mean square error. The values are all in millimeters.

$\mathbf{S_1}$	e_{mean}	σ_e	e_{max}	RMSE
Data-Driven	14.530	25.239	120.237	25.239
Model-Based	18.410	2.724	11.715	7.206
Hybrid	0.686	0.518	1.945	0.861

$\mathbf{S_2}$	e_{mean}	σ_e	e_{max}	RMSE
Data-Driven	18.410	33.052	114.045	37.856
Model-Based	14.089	6.781	26.699	15.739
Hybrid	0.719	0.535	1.908	0.896

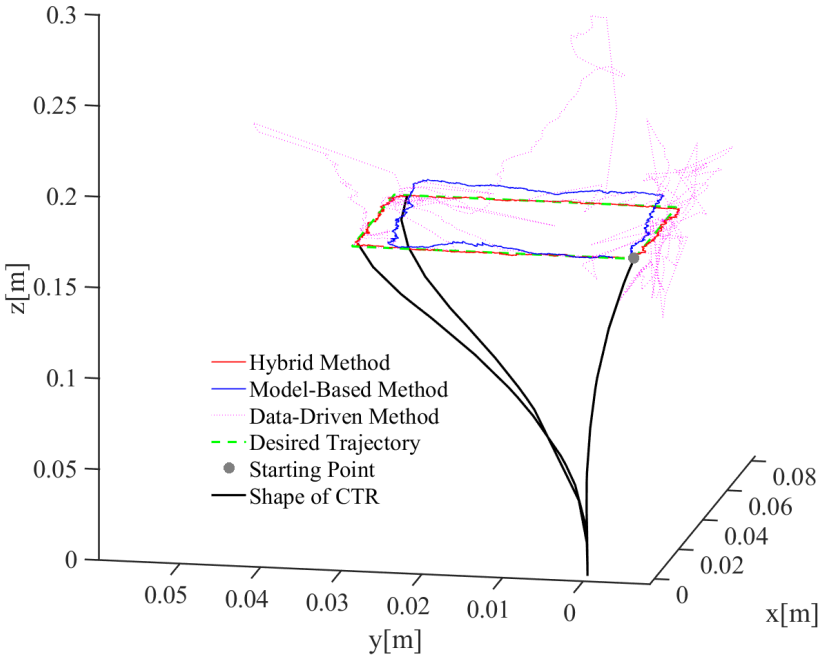
$\mathbf{S_3}$	e_{mean}	σ_e	e_{max}	RMSE
Data-Driven	37.992	48.242	108.476	61.404
Model-Based	5.982	3.180	14.420	6.792
Hybrid	1.105	0.829	4.772	1.382

and is used as a measure of the differences between the actual trajectory of the robot's end-effector, $\boldsymbol{x}$, and the desired trajectory, *i.e.*, $\boldsymbol{x}_d$, for m data points along the robot's tip trajectory.

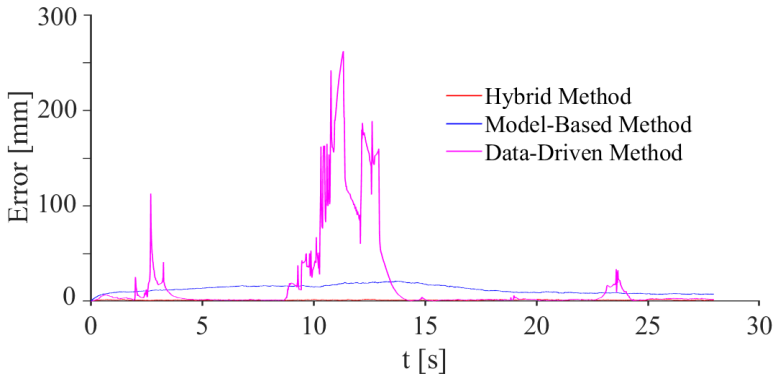
Fig. 4.4(a) shows the actual trajectories of the three compared methods (*i.e.*, data-driven, model-based, and hybrid dual Jacobian) in the 1^{st} scenario (S_1), while Fig. 4.4(b) presents the tracking error versus time. Of note, for these simulations, the pure data-driven approach without any initial training failed to learn the Jacobian and was unable to follow the desired trajectory. This happens because the data-driven method requires a good initial guess to be able to converge. The model-based method struggled to follow the desired trajectory with the unknown changes in the model parameters, while the proposed hybrid method quickly adapted to these changes. The largest errors in this approach appeared when the trajectory changed its direction. The errors with the latter approach are an order of magnitude smaller than with the former ones (as shown in Table 4.2).

In the 2^{nd} scenario (S_2), similar to the previous simulation, the robot was asked to follow a square trajectory with 35 mm base length. Fig. 4.5(a) shows a comparison of the model-based and hybrid methods. Similar to S_1 scenario, the pure data-driven approach without any initial training failed to learn the Jacobian and was not able to follow the desired trajectory. Therefore, due to its large errors, we did not take it into account in the figure. The model-based approach could not also accurately follow the trajectory and obtained RMSE$= 15.7$ mm. As mentioned, this is mainly due to not considering the unknown external forces during deriving the model-based Jacobian. Nevertheless, similar to the previous scenario, the hybrid dual Jacobian

approach is not only able to follow the rectangular trajectory but is much more accurate as well (*i.e.* RMSE$= 0.9$ mm). This is mainly because thy hybrid approach can compensate for the effect of external forces on the deformation behaviour of the robot. Table 4.2 shows that the average error for our proposed dual Jacobian approach is less than 0.72 mm, while it is more than 6.89 mm with the model-based approach, which is not able to adapt to the unknown disturbances.

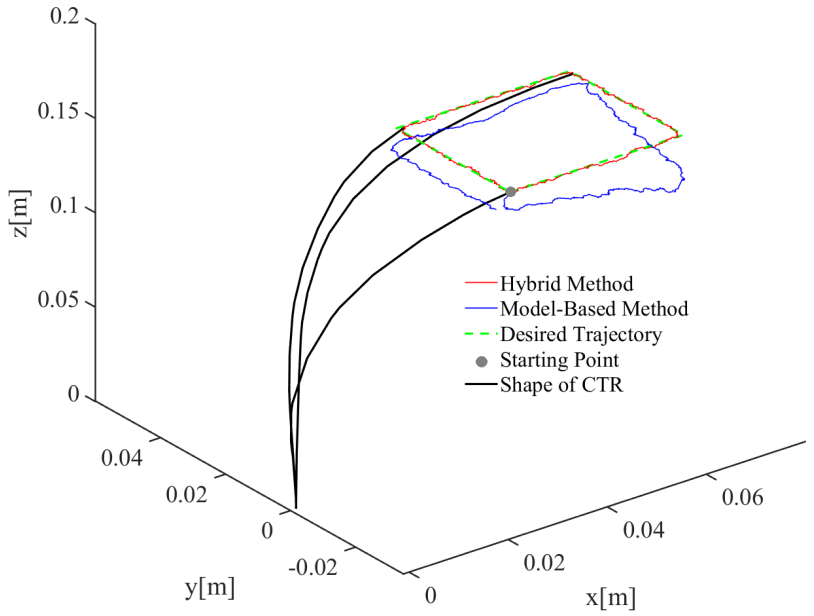


(a)

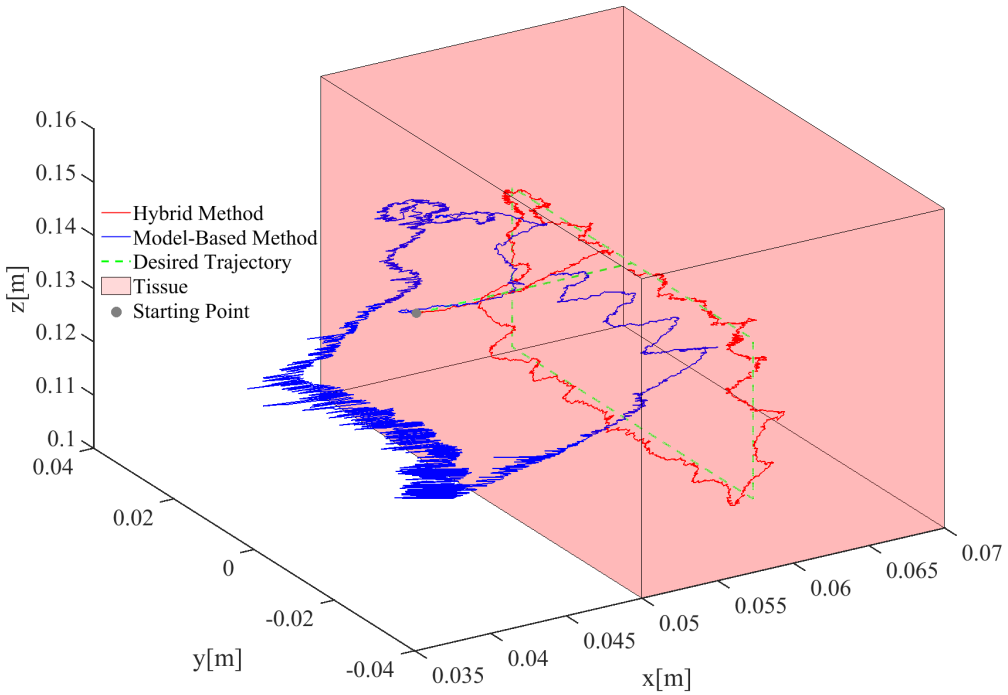


(b)

Figure 4.4: (a) Following a square trajectory using a model with inaccurate parameters. (b) Mean error of the three methods while following a square trajectory using a model with inaccurate parameters.



(a)



(b)

Figure 4.5: (a) Following a square trajectory with a $[0.5, 0.5, 0.5]^T$ N external force on the tip of the robot. (b) The robot is steered to follow a rectangle trajectory while pushing a soft tissue by 1mm in the presence of distributed force along the shape of the robot.

In the 3^{rd} scenario (S_3), the CTR indirectly manipulates particular points on the tissue surface without having prior knowledge about the external contact forces during manipulation. In this simulation, the tissue was placed parallel to the y-z plane at $x = 5$ cm. The external point force was approximately 0.3 N when the tissue was pressed by 1 mm. The results summarized in Table 4.2 demonstrate the superior performance of the proposed hybrid approach. The hybrid approach has some difficulties at the beginning when it reaches the soft tissue and deviates from the desired trajectory when the desired trajectory changes its direction. However, it is able to recover quickly and follow the trajectory with more accuracy compared to the model-based method. As shown in Fig. 4.5(b), the model-based controller struggles to follow the desired trajectory and remains close to the surface of the tissue.

The proposed hybrid approach is able to adapt to uncertainties in the model and in the environment. The hybrid Jacobian proposed in (4.3) always requires some time to learn the effects of uncertainties and disturbances. To investigate the time response of the controller, we performed 16 trials; in each trial, the robot was tasked to follow a trajectory at a velocity between 0.1 to 10 mm/sec with sampling times varying between 15 to 100 milliseconds, in the presence of model uncertainties. Results are summarized in Fig. 4.6. As it can be seen the controller's error linearly increases with respect to tracking velocity. This shows that the Hybrid controller requires sufficient time to adapt to the changes in the feedback signal and is most suitable for autonomous control of the CTR at velocities below 10 mm/sec, which is appropriate for needle-based interventions [89] such as lung biopsy and ablation.

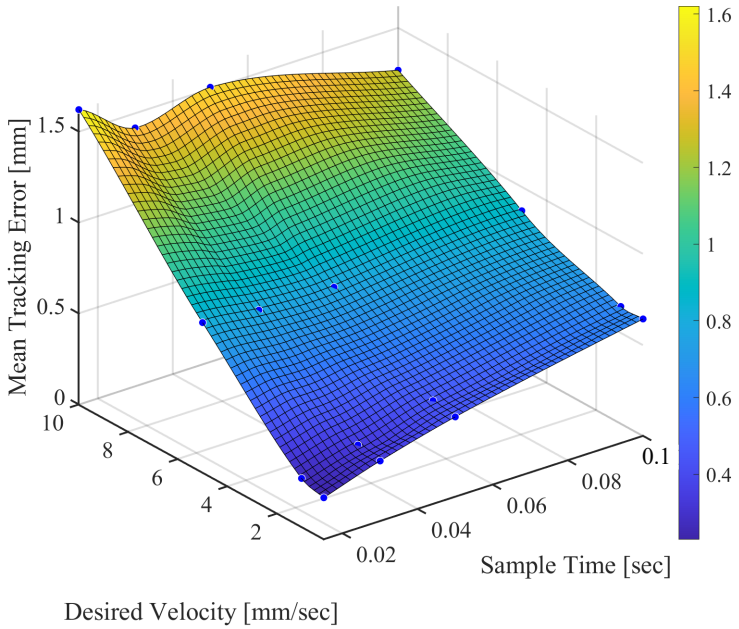


Figure 4.6: Hybrid controller's error with respect to sampling time and desired velocity. The blue points on the plot are the results from the simulations, the surface on the plot was fitted on these points.

4.6 Experimental Verification

4.6.1 Control Architecture Modifications

Here, we refine the hybrid data-driven control architecture developed in the previous sections to control the CTR's end-effector to track a time-varying desired trajectory, $\boldsymbol{x}_d(t)$. To provide precise control of the actual robot, we redefine the data-driven Jacobian of the robot as:

$$\mathbf{J}_H^{k+1} = \mathbf{J}_M^k \mathbb{F} + \left[\mathbf{J}_H^k + \left(\frac{\Delta \boldsymbol{x}^k - \mathbf{J}_H^k \Delta \boldsymbol{q}^k}{(\Delta \boldsymbol{q}^k)^T (\Delta \boldsymbol{q}^k)} (\Delta \boldsymbol{q}^k)^T\right) \boldsymbol{\chi}\right] (1 - \mathbb{F}), \tag{4.12}$$

where $\mathbf{J}_H$ is the data-driven Jacobian, χ is the learning rate, and $\mathbb{F} = \text{diag}([F, F, 0, 0])$, where F is the *Normal Forgetting Function* defined as:

$$F = \exp\left(\frac{-\text{rem}\left(\frac{t}{\Gamma_2}\right)}{\Gamma_1}\right), \tag{4.13}$$

where rem is the remainder function, and Γ_1 and Γ_2 are constant parameters defining the speed of forgetting and remembering, respectively. Figure 4.7 shows the normal forgetting function for various values of Γ_1 and Γ_2. According to this, the translational part of the Jacobian is reloaded based on the model at a specified interval (Γ_2). Between reloads the controller forgets the model-based solution and learns the robot's behaviour based on the known joint input changes and the corresponding tip position displacements. The rate of forgetting is described by (Γ_1). Of note, the reminder function can be implemented with equidistant resets in practice.

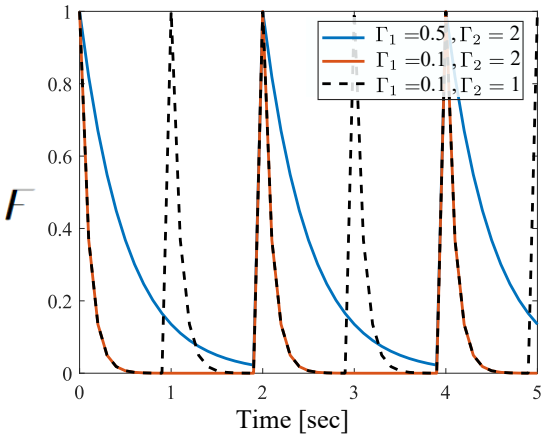


Figure 4.7: Plots of normal forgetting function for various values of Γ_1 and Γ_2. The normal forgetting function reloads the translational part of the Jacobian based on the model at a set interval (Γ_2). Between two reloads the controller forgets the model-based solution at a specified rate (Γ_1) and uses Broyden's update to learn the robot's behaviour.

Compared to previously presented simulation studies, we have improved the Hybrid Jacobian in two ways:

1. We use variable learning rates for translational (β) and rotational (α) actuators. These actuators have different units and different magnitudes. Using variable learning rates allows us to tune the controller more efficiently. To this end, $\boldsymbol{\chi}$ is defined in the following way: $\boldsymbol{\chi} = \text{diag}([\chi_1, \chi_1, \chi_2, \chi_2])$, where χ_1 and χ_2 are learning rates for translational and rotational actuators, respectively.
2. We introduced the normal forgetting function that will enable the controller to employ the information from the model-based Jacobian at fixed time intervals. The reason for this is that the translational inputs of the CTR (β_1 and β_2) have relatively smaller values than the rotational ones. Moreover, CTRs tend to move on curvilinear trajectories, which use rotational inputs more often than translational inputs. Therefore, rotational inputs are often updated more frequently. This will cause the controller to ignore translational inputs or the Hybrid Jacobian to become ill-conditioned. To this end, we use the normal forgetting function to reset the columns of the Jacobian estimated by the controller via the data-driven approach and replace them with the model-based Jacobian. The frequency of forgetting and resetting can be adjusted using Γ_1 and, Γ_2 as shown in Figure 4.7).

Now, we can use the following control law to track an arbitrary desired trajectory.

$$\dot{\boldsymbol{q}}_d = \mathbf{J}_H^{\dagger}[\dot{\boldsymbol{x}}_d + \mathbf{K}(\boldsymbol{x}_d - \boldsymbol{x})], \tag{4.14}$$

where $\boldsymbol{q}_d$ is the desired actuator velocities, $\mathbf{J}^{\dagger}$ is the Moore–Penrose inverse of the hybrid Jacobian matrix. The Moore–Penrose inverse is required since $\mathbf{J}$ is not a square matrix. The proportional gain matrix, $\mathbf{K}$, is a symmetric positive definite matrix.

4.6.2 Experimental Results

To verify the performance of the hybrid controller, a concentric tube robot with two tubes was used with a 5-DOF Electromagnetic tracker (Aurora, NDI) attached to its tip. See Section 3.5 (Fig. 3.9) for more details about the experimental setup.

The controller was verified through the following two cases:

(S_1) *Robot following pre-defined trajectories in free-space:* Two different trajectories were followed by the robot's tip in free-space: (i) A square trajectory with a base length of 20 mm, (ii) a spiral trajectory on the global xy plane.

(S_2) *Robot following pre-defined trajectories under unknown external forces:* In this study, the controller was tested on a square trajectory with a base length of 20 mm while applying a point load of 20 grams to the tip of the robot.

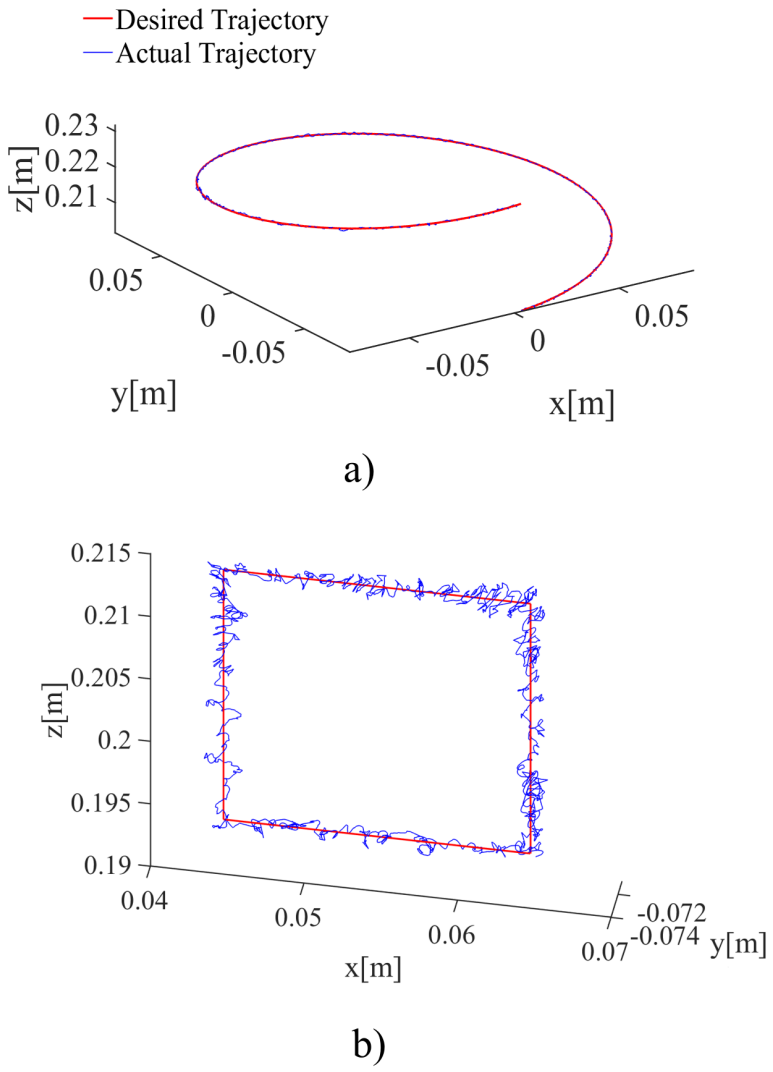


Figure 4.8: Representative experimental results for trajectory tracking in the first scenario (S_1): Robot's tip follows a) a spiral trajectory, b) a square trajectory.

Based on the experimental results, Table 4.2 summarizes the tip position accuracy during deployment. Each experiment was carried out five times. Figure 4.8 illustrates representative results for the first scenario (S_1). Then Figure 4.9 shows the results for the second scenario (S_2) when an unknown weight is added to the tip of the robot. It is important to emphasize that the position accuracy of the electromagnetic sensor is 0.7 RMS (mm) with a maximum error of 1.8 mm. Therefore, the tip position error is mostly due to the sensor's inaccuracy. In addition to this, the figures' axes have different scales, which magnifies the fluctuations in one

direction. Based on this, more advanced controllers including artificial damping may be able to improve the fluctuations but the proposed controller's accuracy is already very good. The results confirm that the hybrid controller can provide precise position control locally, even in the presence of unknown external forces.

Table 4.2: Comparison of tip position error between desired and actual trajectories. e_{mean} is the average error, σ_e is the standard deviation of the error, e_{max} is the maximum error, and RMSE is the root mean square error. The values are all in millimeters. It is important to note that the position accuracy of the electromagnetic sensor is 0.7 RMS (mm) with a maximum error of 1.8 mm.

	e_{mean}	σ_e	e_{max}	RMSE
S_1**: Square**	0.668	0.415	2.550	0.786
S_1**: Spiral**	0.922	0.595	2.85	1.098
S_2**: Square with weight**	0.787	0.389	2.853	0.878

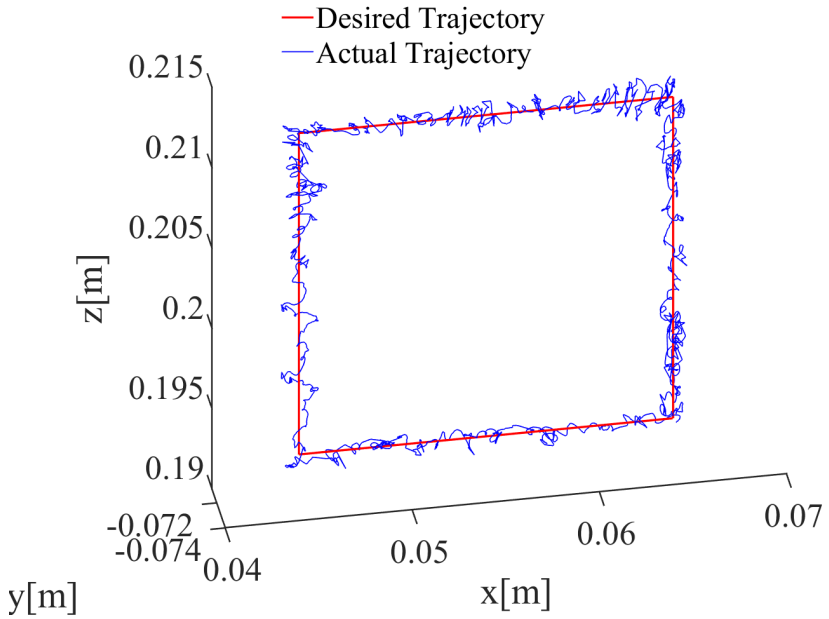


Figure 4.9: Representative experimental results for the second scenario (S_2): Robot's tip follows a square trajectory, while an unknown weight is attached to its tip.

4.7 Conclusion

A hybrid model-based and data-driven controller was described in this chapter. The demonstrated hybrid approach is able to control CTRs experiencing unknown perturbations and external forces. This hybrid approach used an initial Jacobian obtained from the solution of the CTR model and then efficiently combined it with a data-driven approach to update the Jacobian in real-time and estimate the variations in the Jacobian caused by unknown external forces and perturbations. With this approach, the CTR was able to adapt to uncertainties such as unknown external forces (with up to 0.5 N magnitude) and up to 20% error in estimating the model parameters. Various simulation studies demonstrated that the RMSE of the proposed hybrid approach in tracking a rectangular trajectory is about 9 times less than a common model-based controller (*i.e.,* < 1.1 mm). In addition, experiments were conducted on a real CTR to verify the proposed controller. The experiments demonstrated that the proposed controller was capable of controlling the robot's tip despite the applied external forces. Although these results are promising and enable accurate position control of CTRs, the controller still relies on the input provided by the model-based solution and is sensitive to the control parameters. The next chapter utilizes the proposed hybrid controller in possible distal lung sampling applications.

Chapter 5

Robotics-assisted Optical Biopsy

In the previous chapter, a data-driven control approach was introduced to overcome the limitations of model-based control solutions. As a result, the proposed hybrid controller is capable of achieving precise position control if a feasible trajectory is provided. In this chapter, possible applications of CTRs for distal lung sampling are discussed and a motion planner is presented to generate feasible trajectories for specific applications. There are two goals associated with motion planning. First, steering the robot accurately towards the target region while compensating for unknown tissue interaction forces. Second, maintaining the orientation of the robot and the imaging probe to ensure precise imaging of the target region. In order to achieve these goals, a three-phase affordance-aware motion planner is introduced. Affordance in robotics refers to the potential actions that an object such as soft tissue or environment offers to a robot. Experiments have been conducted on phantom tissue to verify the performance of the proposed motion planner for using CTRs for optical endomicroscopy and as a steerable needle.

According to this, Section 5.1 gives an overview of potential applications of CTRs in distal lung sampling. Then, the Affordance-aware Motion Planning is introduced in Section 5.2. This includes the Dexterity Affordance in Section 5.2.2, Stability Affordance in Section 5.2.3 and Constrained Motion Affordance in Section 5.2.4. Following this, a 3-phase motion planner is introduced in Section 5.3. Finally, the experimental results are presented in Section 5.4.

The proposed motion planner for robotic-assisted optical biopsy was also submitted to the IEEE Transactions on Medical Robotics and Bionics(T-MRB) as follows:

B Thamo, V Voulgaridou, H Wood, J Stone, K Dhaliwal and M Khadem", Towards Robotics-Assisted Endomicroscopy in Percutaneous Needle-based Interventions", *IEEE Transactions on Medical Robotics and Bionics* 2023 (T-MRB 2023, submitted).

5.1 Introduction

In the past decade, advances in fibre optics, light sources, detectors, and molecular biology have led to the development of several novel methods for in vivo endomicroscopy. The term endomicroscopy refers to methods that use the properties of light to enable the operator to make an instant in-vivo diagnosis, previously only possible by using histological or cytological analysis. According to this, a number of promising imaging techniques has emerged, including Fluorescence Imaging, Optical Coherence Tomography, Confocal Endomicroscopy, and Surface-Enhanced Raman Spectroscopy. These technologies have shown to be promising tools for tissue characterization compared to traditional biopsy. Instead of taking a number of biopsies and examining the samples with histology, an optical probe can be inserted into the patient to scan a target region. Modalities of endomicroscopy have been combined with surgical robots to provide high-resolution images of the tissue in real-time. A review of various applications is provided in [90]. However, the application of these technologies is limited to surface or near-surface imaging due to the lack of steerable technology that can (i) guide the probe inside the tissue, and (ii) provide precise control over the distal scanning motion. In this chapter, we detail the proof of concept for a robotic platform capable of accurate and repeatable endomicroscopic imaging in needle-based interventions (Figure 5.1). In needle-based interventions, a physician inserts a needle through the skin to reach different internal organs, such as prostate, breast, lung, and liver to perform a biopsy or target cancerous lesions with needle-based ablation procedures to burn or freeze malignant cells. Percutaneous lung biopsy and prostate biopsy or ablation are two examples shown in Figure 5.2.

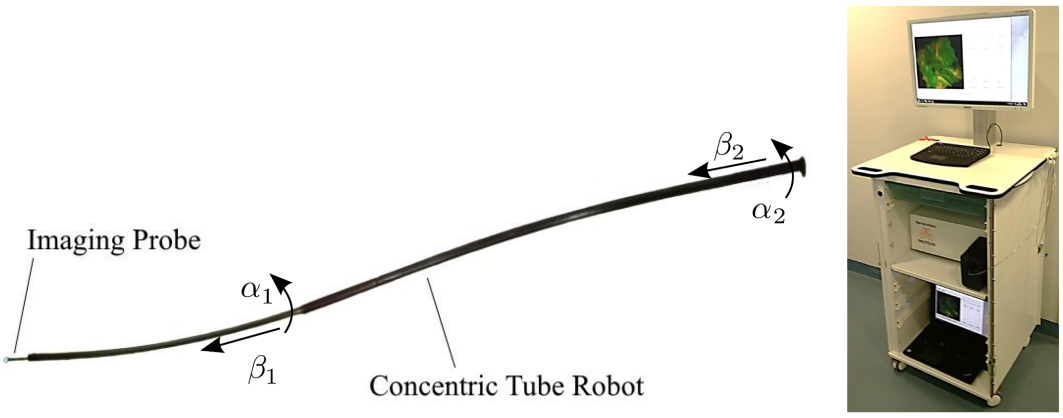
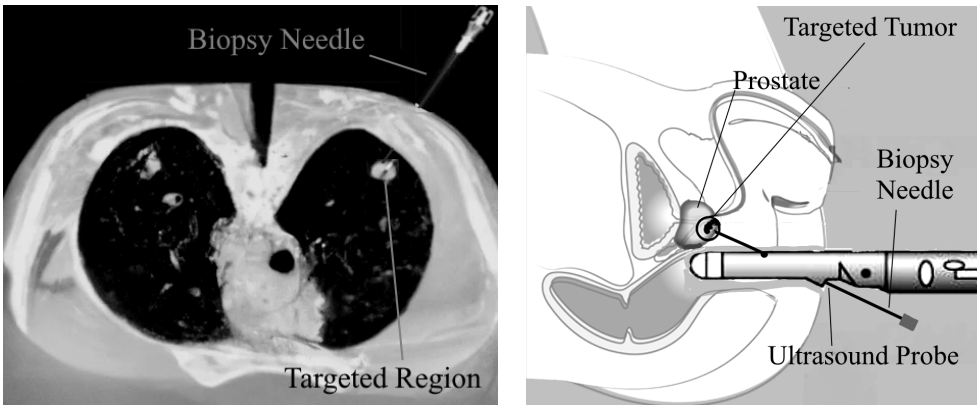


Figure 5.1: (a) A CTR used as a steerable needle deploying a fluorescence imaging probe connected to the Versicoulor fluoroscopy imaging unit. The CTR's shape and length can be controlled by axially rotating (α_1, α_2) or translating (β_1, β_2) the pre-curved tubes. (b) The medically approved Versicolour unit. Figure (b) is from [1] with permission of the authors.

To fully take advantage of the endomicroscopic technology, we propose a robotic platform that consists of a continuum robot, namely, a CTR and a bespoke fluorescence endomicroscopic imaging platform called *Versicolour*, first introduced in [1]. Versicolour is a sensitive and modular three-colour fluorescence endomicroscopic imaging platform spanning the visible to near-infrared (NIR) range. Versicolour has been clinically translated into patients with pulmonary disease to delineate healthy, cancerous, and fibrotic tissue autofluorescent structures [1]. Using this technology, the proposed robotic platform is experimentally validated for automated endomicroscopic imaging during needle-based interventions. This technology can offer accurate in-vivo real-time imaging of targeted regions during needle-based interventions to (i) guide the biopsy, (ii) monitor the progression of a disease and assess the effectiveness of an experimental treatment, (iii) unify the diagnosis and the treatment of diseases or cancerous cells by providing real-time diagnosis followed by the delivery of targeted therapies or ablation. To achieve the above-mentioned goals, a CTR is used as a steerable needle to steer a flexible endomicroscopic probe deep inside the tissue. In this regard, the main challenges are (i) the accurate steering of the robot towards the target region while compensating for unknown tissue interaction forces, (ii) controlling the orientation of the robot and the imaging probe to guarantee precise imaging of the target region. To this end, we propose a motion planning algorithm to calculate the feasible target regions for these procedures. Then, the initial configuration of the robot is calculated to reach the target with the desired tip orientation. Finally, the hybrid controller described in Chapter 4 is applied to steer the robot on a pre-planned path while compensating for unknown tissue effects. Experiments are performed to validate the robotic platform in two scenarios simulating clinical cases. To the best of our knowledge, this is the first robotic platform capable of automated deep tissue endomicroscopic imaging.



(a) (b)

Figure 5.2: Examples of percutaneous needle-based Interventions. (a) Needle-based lung biopsy. A CT Scan of the chest is shown during the procedure. (b) Transrectal prostate biopsy under ultrasound guidance.

In this chapter, we develop and experimentally validate the robotic platform for automated endomicroscopic imaging during needle-based interventions. We evaluate the robot in two scenarios: (i) scanning a phantom tissue region, while searching for anomalies in fluorescent images, simulating percutaneous lung biopsy; (ii) accurately reaching a target deep inside a phantom tissue at a relative depth of 50 mm (Figure 5.3).

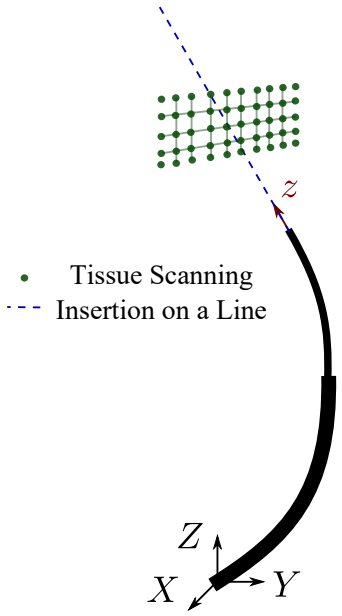


Figure 5.3: Two applications of the proposed motion planner, namely, insertion in a straight line into the tissue and scanning the tissue on a mesh grid.

5.2 Affordance-aware Motion Planning

Here, we propose a motion planning algorithm that accepts the entry point and the desired path to a specific procedure in a global coordinate frame estimated from pre-operative medical images as input and finds (i) the desired robot configuration at the entry point for a specific task, (ii) a trajectory from the robot's initial configuration to the desired entry point, and (iii) a stable and feasible trajectory from the entry point towards the target.

In order, to take endomicroscopic images of the target region inside the tissue, the generated plan should satisfy the following requirements:

1. The generated plan should guide the robot toward the target in a way that the robot's tip is roughly perpendicular to the tissue's surface. This requirement ensures that the endomicroscopic probe is capable of capturing high-quality images.

2. The generated plan should minimise the interaction forces between the robot and the tissue. As the robot cuts through the tissue, the robot will bend as a result of the interaction forces and deviate from the desired trajectory. These forces are unknown and hard to estimate in heterogeneous tissue [91]. Therefore, the planner should aim to find a trajectory where interaction forces are minimal. Later, the unknown external forces can be handled by the controller as disturbances.
3. The planned trajectory should be stable and avoid singularities. CTRs exhibit elastic instabilities due to torsional elastic energy storage in the tubes. An instability occurs when this energy is rapidly released and the robot "snaps" to a new configuration [92]. The generated plan should avoid such instabilities.

In line with the requirements to generate a motion plan for CTRs for deep tissue endomicroscopy, we employ the concept of affordance to quantify the aforementioned requirements in a way that can be considered by a motion planner. Essentially, affordance describes the relationship between an object and the capabilities of the robot to interact with it. Furthermore, affordance provides guidance for robot design and can improve the efficiency and effectiveness of robot-object interactions [93]. Accordingly, in the following sections, we characterize the robot's affordance for deep tissue endomicroscopy.

5.2.1 CTR Affordance for Endomicroscopy

Based on the defined requirements, we intend to quantify the robot's affordance for (i) moving perpendicular to the tissue, (ii) cutting with minimal force, and (iii) avoiding singularities.

To achieve this, we use the concept of manipulability for continuum robots introduced in [94]. Velocity manipulability is a measure of a robot's ability to produce changes in its velocity by altering its joint inputs. It is a metric that assesses the controllability of the robot's motion. In accordance with this, CTR manipulability can be estimated using the robot's kinematic model, which is detailed in Section 3.1, and can be summarised as

$$\boldsymbol{x} = f(\boldsymbol{q}(t), \boldsymbol{g}(s), \boldsymbol{u}(s)), \tag{5.1}$$

where x is the Cartesian coordinates of the robot's end-effector, t and s denote time and robot's arc length, $\boldsymbol{q}(t) = [\beta_1,\ \beta_2,\ \ \alpha_1,\ \alpha_2,]^T$ denotes the actuation values shown in Figure 5.1, $\boldsymbol{u}(s)$ is the curvature of the robot's backbone, and, $\mathbf{g}(s) \in \mathrm{SE}(3)$ is a homogeneous transformation defining the location and orientation of the robot's backbone in task space and can be defined as

$$\mathbf{g}(s) = \begin{bmatrix} \mathbf{R}(s) & \boldsymbol{r}(s) \\ \mathbf{0}_{3\times 1} & 1 \end{bmatrix} \tag{5.2}$$

where $\boldsymbol{r}(s)$ is the arc-length parametrized shape of the robot and $\mathbf{R} \in \mathrm{SO}(3)$ is a rotation matrix at every arc-length s. Using the model, we can define the Jacobian matrix, $\mathbf{J}$, that maps the joint velocities, $\dot{\boldsymbol{q}} \in \mathbb{R}^4$, to the robot end-effector velocity, $\dot{\boldsymbol{x}} \in \mathbb{R}^3$ as

$$\dot{\boldsymbol{x}} = \mathbf{J}\dot{\boldsymbol{q}}. \tag{5.3}$$

The Jacobian can be numerically estimated by solving the model in (5.1):

$$\mathbf{J} = \frac{\Delta \boldsymbol{x}}{\Delta \boldsymbol{q}} = \begin{bmatrix} \frac{\boldsymbol{x}^T\left(\boldsymbol{q}+\frac{\Delta q_1}{2}\boldsymbol{e}_1\right)-\boldsymbol{x}^T\left(\boldsymbol{q}-\frac{\Delta q_1}{2}\boldsymbol{e}_1\right)}{\Delta \boldsymbol{q}_1} \\ \vdots \\ \frac{\boldsymbol{x}^T\left(\boldsymbol{q}+\frac{\Delta q_4}{2}\boldsymbol{e}_4\right)-\boldsymbol{x}^T\left(\boldsymbol{q}-\frac{\Delta q_4}{2}\boldsymbol{e}_4\right)}{\Delta \boldsymbol{q}_4} \end{bmatrix}^T. \tag{5.4}$$

Where $\boldsymbol{e}_i$ is the i^{th} unit vector of the canonical basis of joint space. For a two-tubed CTR, the Jacobian is a 3×4 matrix.

Now, we can use the Jacobian to estimate the robot's velocity manipulability ellipsoid (VME). VME represents the direction in which the robot is capable of moving in a given configuration. Furthermore, the VME maps a unit sphere in joint space to an ellipsoid in task-space and can be estimated by taking the Singular Value Decomposition (SVD) of the Jacobian as

$$\mathbf{J} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T, \tag{5.5}$$

where the diagonal entries of $\boldsymbol{\Sigma} = diag(\sigma_1..\sigma_n)$ are singular values uniquely determined by $\mathbf{J}$ and $\sigma_1 > \sigma_2 > \sigma_i > \sigma_n$. The columns of $\mathbf{U}$ and the columns of $\mathbf{V}$ are called left-singular vectors and right-singular vectors of $\mathbf{J}$, respectively. By using σ_i and $\boldsymbol{v}_i$, the VME can be spanned where $\boldsymbol{v}_i$ is the i^{th} column of $\mathbf{V}$. Moreover, $\boldsymbol{v}_i$ is the i^{th} principal axis vector of the VME with a magnitude equal to σ_i.

Additionally, one can develop the Unified Force Manipulability Ellipsoid (UME) defined in [94] to quantify the direction of robot motion in response to a unit force/torque applied to its tip. The schematic of the two ellipsoids is shown in Figure 5.5(a). Of note, as illustrated in [94] UME's major axis is approximately parallel to the VME's minor axis. Therefore, the direction where the robot has the highest ability to produce changes in its velocity is the same as the direction where it retains maximum stiffness (*i.e.*, minor axis of UME). Now, by leveraging some of the features of the VME, we propose three variables to quantify CTR affordance for endomicroscopic tasks, namely, *Dexterity Affordance* (ψ), *Stability Affordance* (c), and *Constrained Motion Affordance* (Γ).

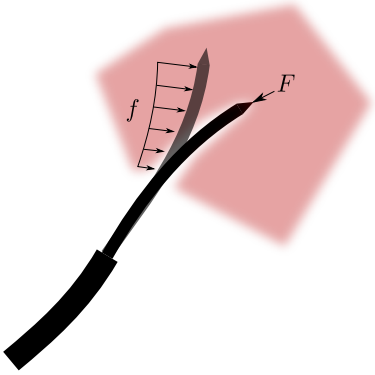


Figure 5.4: A schematic of the CTR advancing through soft tissue. As the robot proceeds, it cuts tissue with its symmetric sharp inner tube. As a result, external cutting forces are applied to the robot's tip ($\boldsymbol{F}$). When the robot deviates from the cutting path, which is in the direction of its local z-axis, it compresses tissue and distributed external forces ($\boldsymbol{f}$) are applied to its backbone.

5.2.2 Dexterity Affordance

Here, we define variable ψ to quantify how well the robot can move instantaneously through soft tissue for cutting and taking endomicroscopic images. As the CTR advances through soft tissue, cutting forces are applied to its tip in the opposite direction of the tip's local z-axis [91](Figure 5.4). In a configuration, where the principal axis of the VME with the largest singular value ($\boldsymbol{v_1}$) is aligned with the local z-axis of the robot's tip, the CTR can manipulate its velocity along the local z-direction with less joint efforts (left image in Figure 5.5(a)). Additionally, based on the fact that the UME is approximately perpendicular to the VME, the robot retains high stiffness against cutting forces (CTR will bend less under axial forces). Therefore, this is the optimal case for cutting through tissue. Furthermore, the robot advances along its local z-axis and will reach the desired targets perpendicularly. Thus, it would be able to capture high-quality endomicroscopic images of the target. To this end, we define the Dexterity Affordance ψ as the angle between the tip's local z-axis and the major axis of the velocity manipulability:

$$\psi = \arccos(\boldsymbol{z} \cdot \boldsymbol{v_1}), \tag{5.6}$$

where v_1 is the first column of $\mathbf{V}$ in (5.5) and $\boldsymbol{z}$ is the last column of matrix $\mathbf{R}(s)$ at the robot's tip, which is estimated by solving the kinematic model of the robot in (5.1). Note that both vectors $\boldsymbol{z}$ and $\boldsymbol{v}_1$ are unit vectors.

It is important to emphasize that a configuration with small ψ has more dexterity in moving along its local z-axis making it more capable of cutting through tissue and acquiring high-quality images by pressing the endomicroscopic probe perpendicularly to the tissue. This is demonstrated in Figure 5.5(b). Of note, we only consider the columns of the Jacobian which

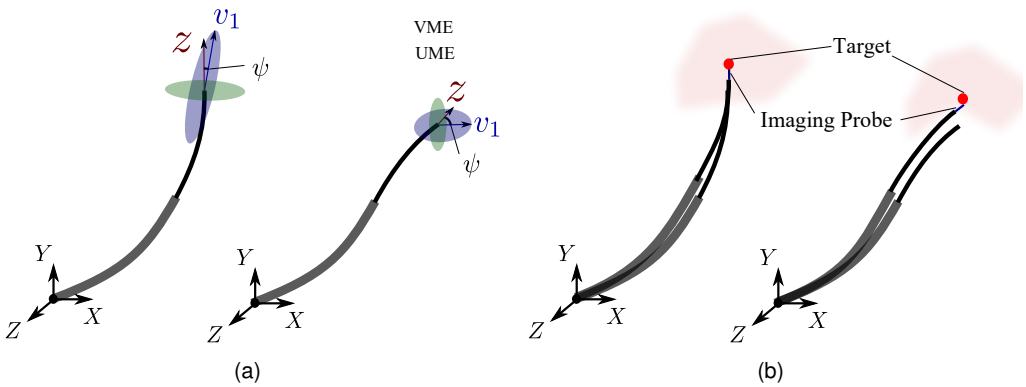


Figure 5.5: (a) A schematic of the velocity manipulability ellipsoid and unified force manipulability ellipsoid of a CTR in two different configurations. The major axis of the VME (v_i), the local z-axis of the robot at its tip (z), and the dexterity affordance (ψ) are shown as well. (b) A schematic of the robot moving along its principal axis of velocity manipulability ellipsoid in two different configurations. When the VME's principal axis is aligned with the robot's local z-axis, the robot is capable of reaching the desired target perpendicularly. This is more suitable for endomicroscopic imaging.

correspond to translational actuators (*i.e.*, β_1 and β_2). The reason is that (i) intuitively we know that the robot generates axial motions using its translational actuators, and (ii) rotational and translational actuators have different units. This can lead to inconsistent results when calculating the manipulability of the robot, as the units used in the calculation will affect the magnitude of the singular values.

5.2.3 Stability Affordance

CTRs can become unstable or "snap" when the robot's forward kinematics loses its uniqueness and can have multiple equilibriums. Snapping is a mechanical instability caused by the rapid release of the elastic potential energy that is accumulated due to the bending and twisting of the tubes [92]. Snapping, the unexpected swift change in the CTR's configuration complicates the process of performing tasks smoothly and can harm sensitive tissues in the proximity of the robot. In order to characterize the solution of robot kinematics and its stability, we introduce the robot's stability affordance (c) as the inverse of the robot's Jacobian's condition number:

$$c = \frac{1}{\|\mathbf{J}\|\|\mathbf{J}^{-1}\|} = \frac{\sigma_n}{\sigma_1}. \tag{5.7}$$

According to this, c is calculated by dividing the most significant singular value of $\mathbf{J}$ by the least significant singular value of $\mathbf{J}$. The Jacobian matrix becomes singular as the robot kinematics becomes unstable. Consequently, c increases towards infinity. By minimizing the proposed stability affordance, one can avoid such scenarios.

The proposed Dexterity Affordance and Stability Affordance quantify the robot's motion for a given configuration but cannot provide information on the transient behaviour of the CTR. To this end, we introduce another variable, namely, the *Constrained Motion Affordance*.

5.2.4 Constrained Motion Affordance

As discussed, when the robot cuts through tissue or scans the tissue for endomicroscopic imaging, its tip is required to move along its local z-axis. The Constrained Motion Affordance aims to quantify the robot's ability in moving along its local z-direction from one point to another. Consequently, we introduce $\Gamma = [\gamma_1, \gamma_2]$, where γ_1 is the relative angle between the tip displacement in the next time step and the tip's local z-axis in the previous time step. γ_2 is the angle between the robot's local z-axis in the next time step and the local z-axis of robot's initial configuration at t_0:

$$\gamma_1 = arccos\left(\frac{(\boldsymbol{x}_{t_k} - \boldsymbol{x}_{t_{k-1}}) \cdot (\boldsymbol{z}_{t_k})}{\|(\boldsymbol{x}_{t_k} - \boldsymbol{x}_{t_{k-1}}) \cdot (\boldsymbol{z}_{t_k})\|}\right) \tag{5.8a}$$

$$\gamma_2 = arccos(z_{t_0} \cdot z_{t_{k+1}}) \tag{5.8b}$$

Figure 5.6 demonstrates these variables, where the robot is shown in several configurations. Assuming that it has moved from point A to point B at time t_k, now we need to find the next feasible point to move to at time t_{k+1}. The robot should move on a straight line to cut through the tissue with its tip, and should avoid lateral movements as much as possible. Furthermore, the robot should approach the target perpendicularly. Accordingly, γ_1 characterizes the robot's motion along a straight path, while γ_2 quantifies the tip's orientation during the motion. Based on this the following scenarios are possible:

1. *Optimal Scenario*: The robot moves on a relatively straight line and approaches the next point perpendicularly creating optimal conditions for endomicroscopic imaging. This scenario is shown in Figure 5.6(b), where both γ_1 and γ_2 are relatively small.
2. *Lateral Movement*: A movement with a relatively large γ_1 results a lateral robot movement. For instance, moving to point D from point B results in a small γ_2. However, as shown in Figure 5.6(c), moving to point D also results in a relatively large γ_1. In this case, the robot has moved laterally. This will compress the surrounding tissue, resulting in relatively large interaction forces.
3. *Tip Orientation Misalignment* The robot's tip is not perpendicular to the target point. Figure 5.6(d) shows a scenario where γ_1 is small and the robot's trajectory is straight, however γ_2 is relatively large. In this case, the robot's tip will not be able to deploy the endomicroscopic probe into the target region perpendicularly.

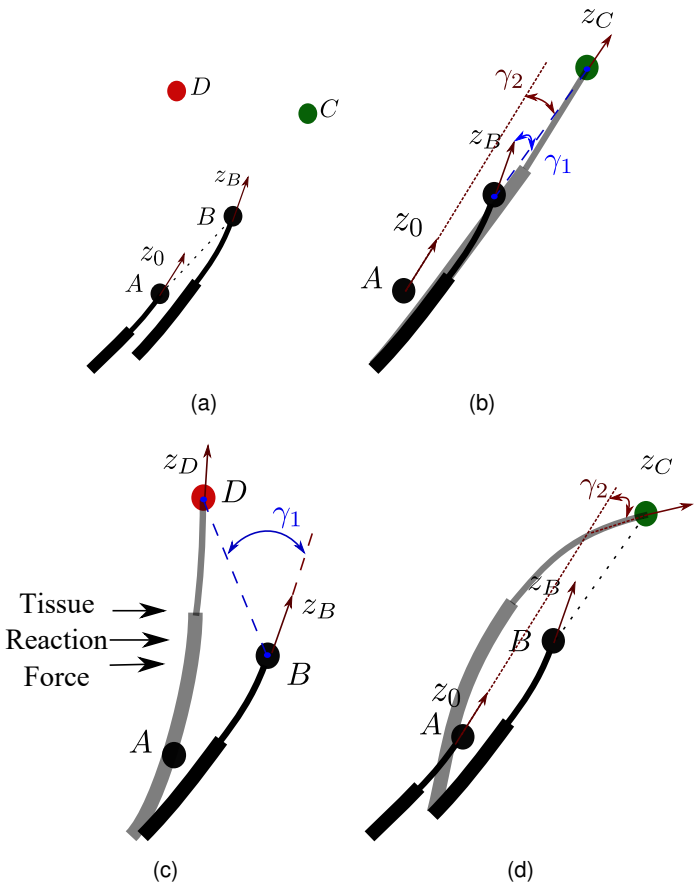


Figure 5.6: A schematic of a CTR moving in 2D space. (a) The robot has reached point B from point A. (b) The robot moves to point C on a relatively straight line (small γ_1) and approaches point C perpendicularly (small γ_2) (c) The robot approaches a target perpendicularly (small γ_2), however, it is making a lateral movement (large γ_1). (d) The robot moves on a straight line (small γ_1) but the robot's tip is not perpendicular to the target point (large γ_2).

In the next section, we use a motion planning algorithm that leverages the developed affordance variables to estimate optimal paths for robotic-assisted endomicroscopy.

5.3 3-Phase Motion Planner

Based on the previously defined robot affordance measures, we propose a motion planning algorithm comprised of 3 different planning phases. The motion planner accepts the desired target location acquired from pre-operative images and employs the affordance variables discussed in the previous section to (i) estimate the robot's entry point into the tissue, (ii) find the optimal trajectory from the entry point to the target to perform endomicroscopic imaging, (iii) find a trajectory that takes the robot to the entry point from a given starting configuration.

5.3.1 Phase 1: Identify ideal starting configurations

The first motion planning phase includes an exhaustive search to find all the configurations in which the robot is stable and has maximum capability in moving along its local z-direction for tissue cutting and scanning. This can be achieved by optimizing the stability affordance variable c and the dexterity affordance ψ. To this end, we generated all possible feasible configurations ($\boldsymbol{q}$) between the robot's joint limits with a 1 mm resolution for translational joint inputs and a 5 degree resolution for rotational joint inputs. Based on the stability affordance and the dexterity affordance measures, we only accepted configurations where $c \leq \varepsilon_1$ and $\psi \leq \varepsilon_2$, where ε_1 and ε_2 are hyperparameters and should be small enough to ensure the robot's stability and dexterity. Based on this, Figure 5.7 shows the results of the 1^{st} phase of the motion planner. The motion planner employs a CTR with parameters given in Chapter 3 (Table 3.3), while ε_1 and ε_2 were selected to be 44.7 and 5°, respectively. ε_1 was selected as the mean of the stability affordance variable c for all generated configurations, this way the majority of the configurations where the robot is unstable are eliminated. ε_2 should ideally be 0, however, this significantly limits the number of feasible configurations. Therefore, a small value of 5° was selected to ensure sufficient configurations are generated in the 1^{st} phase.

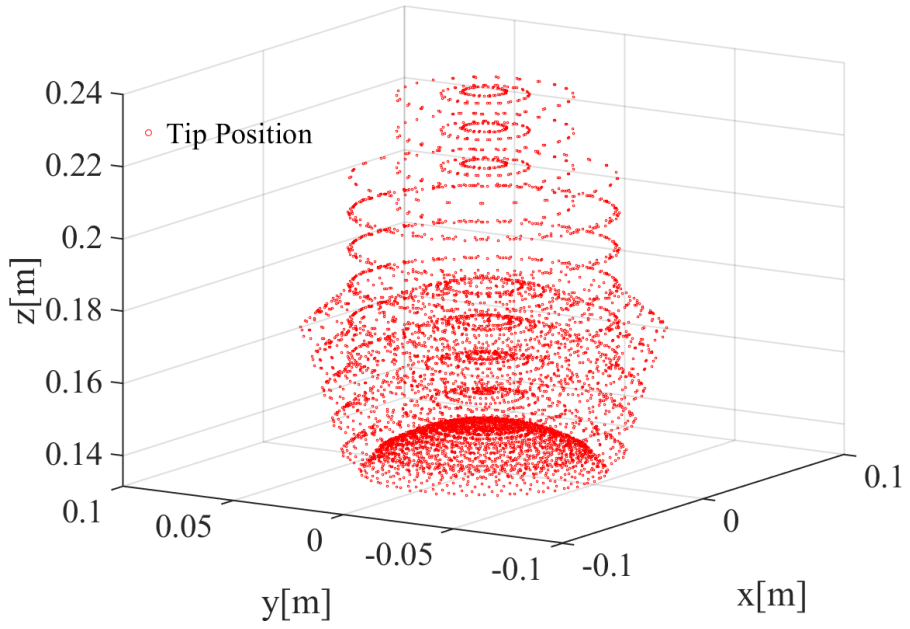


Figure 5.7: Results of the 1^{st} phase of the motion planner. It shows all those tip positions where the dexterity and stability affordance are optimal.

5.3.2 Phase 2: Select the best starting configuration from Phase 1 and generate a feasible trajectory for a given task

In the second phase, a modified RRT* algorithm is applied to select a configuration where the robot can perform insertion into soft tissue on a straight line and scan a region of tissue. The algorithm accepts the results of the previous phase as inputs and runs the planner for each point from phase 1. Based on this, the algorithm is looking for the following factors:

1. *Tissue Insertion*: For tissue insertion, the goal is to find a series of points along a straight line aligned with the local z-axis of the robot's tip (Figure 5.8(a)).
2. *Grid Scan*: For tissue scanning, the goal is to find points along a grid with an offset from the robot's tip position in its local z-direction (Figure 5.8(b)).

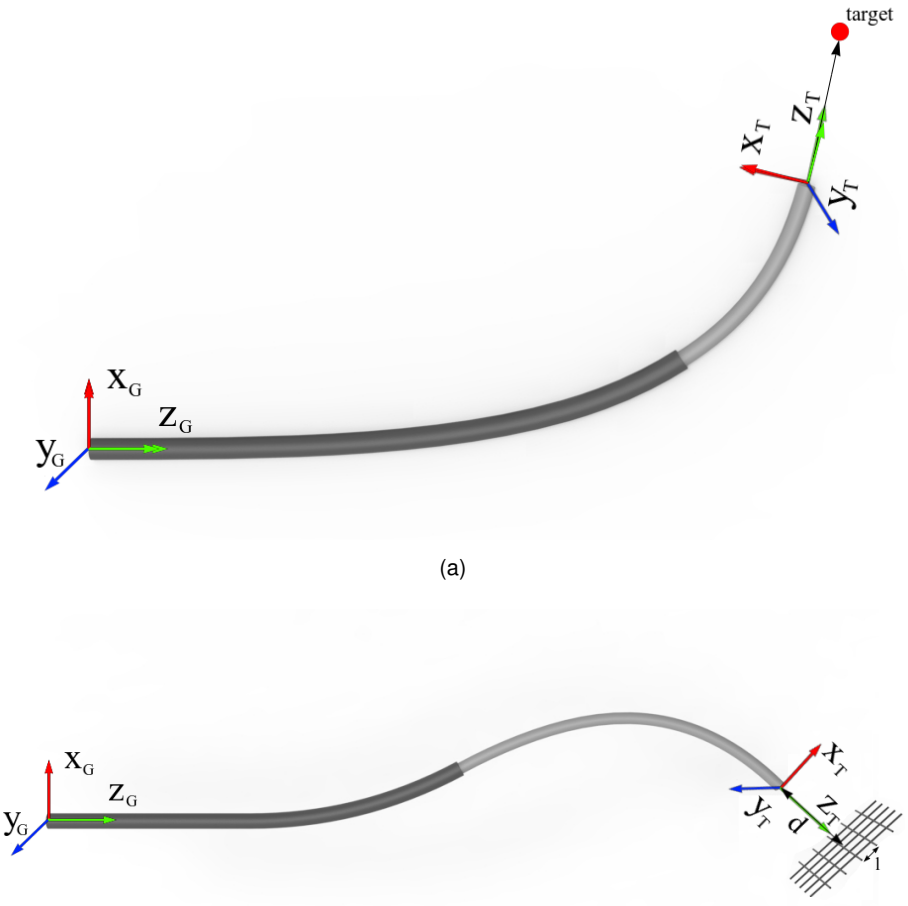


Figure 5.8: (a) First motion planning goal is to find a configuration which allows the robot's tip to move in the tip's local z-direction for 50 mm to reach the $target$. (b) The second motion planning goal is to find a configuration where the robot's tip can reach multiple points along a grid, d is the distance between the tip of the robot and the grid's plane, while l is the distance between two points on the grid. x_G,y_G,z_G show the global coordinate system, while x_T,y_T,z_T indicates the tip coordinate system.

To achieve these motion planning goals, a modified version of RRT* algorithm [52] is applied to generate a motion plan containing a sequence of stable configurations $\Pi = (\boldsymbol{q}_{start}\boldsymbol{q}_1..\boldsymbol{q}_k\boldsymbol{q}_{goal})$ between starting position $\boldsymbol{x}_{start}$ with the corresponding $\boldsymbol{q}_{start}$ initial configuration and goal position $\boldsymbol{x}_{goal}$ with the corresponding goal configuration $\boldsymbol{q}_{goal}$. $\boldsymbol{q}_1..\boldsymbol{q}_k$ are the generated configurations between initial and goal configurations, while $\boldsymbol{x_i} \in \mathbb{R}^3$ is the corresponding tip position of $\boldsymbol{q_i} \in \mathbb{R}^4$ configuration of the robot.

For finding a plan Π, we first build a tree $\boldsymbol{\Gamma} = (\boldsymbol{q}_1..\boldsymbol{q}_k)$, where each element (node) of $\boldsymbol{\Gamma}$ is a valid configuration based on the stability affordance and constrained motion affordance criteria. The algorithm has the following steps:

1. *Generate new node:* Generate a random configuration $\boldsymbol{q}_{rand}$ in configuration space (C-space). The random configuration is generated within the joints' limits. Joint limits are defined as:
   - The translational joint inputs are required to be within the following range: $0 <= \beta_i <= l_i$, where β_i is the translational input of the i^{th} tube and l_i is the length of the i^{th} tube.
   - *Relative limits:* The tubes cannot pass each other, therefore: $\beta_1 < \beta_2$.
2. *Find nearest node:* Find the nearest tip position $\boldsymbol{x}_{nearest}$ in Γ to the generated configuration $\boldsymbol{x}_{rand}$ in work space (W-space).
3. *Contraction:* Create a new configuration $\boldsymbol{q}_{new}$ by moving $\boldsymbol{q}_{rand}$ within a distance δ to $\boldsymbol{q}_{nearest}$:
$$\boldsymbol{q}_{new} = \boldsymbol{q}_{nearest} + \delta \times w \times (\boldsymbol{q}_{rand} - \boldsymbol{q}_{nearest}), \tag{5.9}$$
   where w is a 4×1 weighting vector.
4. *Finding new parent:* After $\boldsymbol{q}_{rand}$ was moved closer to $\boldsymbol{q}_{nearest}$, find a parent node in $\boldsymbol{\Gamma}$ within a distance δ to $\boldsymbol{q}_{new}$ where the cost of reaching $\boldsymbol{q}_{new}$ is lower through the new parent node $\boldsymbol{q}_{parent}$ than through $\boldsymbol{q}_{nearest}$.
5. *Obstacle check:* Check that there is no virtual obstacle within $\boldsymbol{\delta}_{obstalce}$ to $\boldsymbol{q}_{new}$. In such a case, return to the first step.
6. *Stability check:* Check if $\boldsymbol{q}_{new}$ is a feasible solution according to the stability affordance and constrained motion affordance criteria. If $\boldsymbol{q}_{new}$ is unstable, then add it to $\boldsymbol{O}$ as a virtual obstacle and go back to the first step.
7. *Adding new node:* If $\boldsymbol{q}_{new}$ is stable and obstacle free then add $\boldsymbol{q}_{new}$ to $\boldsymbol{\Gamma}$ and do the next step. Otherwise, go back to the first step.
8. *Rewiring:* After $\boldsymbol{q}_{new}$ was added to the tree, we go through the nodes in $\boldsymbol{\Gamma}$ and check if $\boldsymbol{q}_{new}$ lowers the cost of reaching any of the existing nodes.
9. The above-mentioned steps are repeated until the specified goal is reached or the maximum number of iterations is reached. If the goal is not achieved, this process is repeated for another point from the list of entry points generated in phase 1.

The proposed planner is a modified version of RRT* with the following two differences:

(1) We consider configurations where the stability and constrained motion affordances are not optimal as virtual obstacles, and avoid traversing through these configurations. $\boldsymbol{O} = (\boldsymbol{O}_1..\boldsymbol{O}_t)$ where $\boldsymbol{O}_i \in \mathbb{R}^4$ is the list of all configuration where $C \leq \varepsilon_1$ and $\gamma_2 \leq \varepsilon_3$, where ε_1 are ε_3 hyperparameters.

(2) In general, applying the RRT* to CTRs is challenging due to the different joint types (translational and rotational) with different units and magnitudes. As a result, step 2 and 3 in the algorithm (*i.e.*, finding the distance between two configurations in C-space) is challenging. Therefore, the RRT* will almost certainly be biased toward using one type of actuation (rotational or translational), preventing it from exploring W-space. In this regard, joint weights (w in (5.9)) are applied in step 3 to prevent biasing toward rotational or translational movements. In order to estimate w for a given configuration ($\boldsymbol{q}_{nearest}$), each joint i is perturbed by a small value ($\delta\alpha$ and $\delta\beta$). Next, the tip displacement for each perturbation is recorded (δx_i). Finally, w is calculated as

$$w = \left[\frac{\delta x_{\text{max}}}{\delta x_1}, \frac{\delta x_{\text{max}}}{\delta x_2}, \frac{\delta x_{\text{max}}}{\delta x_3}, \frac{\delta x_{\text{max}}}{\delta x_4} \right]^T, \tag{5.10}$$

where $\delta x_{\text{max}} = \max\{\delta x_i\}$, for $i = 1, \cdot\cdot, 4$. w aims to normalize joint displacement during contraction to ensure that a unit change in each joint results in similar tip displacement, thus avoiding bias toward one type of motion.

Figure 5.9 shows the result of the first two phases of the motion planner for two scenarios: (i) moving on a straight line (green), (ii) scanning the tissue surface (blue). For insertion on a line, the goal is to follow a 50 mm long straight line into the tissue. For achieving this the hyperparameters were selected as $\varepsilon_1 = 44.7$ and $\varepsilon_3 = 15$ where ε_1 was selected as the mean of the stability affordance variable c for all generated configurations. For tissue scanning, the goal was to track a mesh grid normal to the robot's entry point with a 25 mm offset from the tip position in the local z-direction. The mesh contains a 5 by 5 grid with 10 mm distance between each point. The hyperparameters for this scenario were selected as $\varepsilon_1 = 44.7$ and $\varepsilon_3 = 60$. Thresholds for constrained motion affordance were relaxed as for tissue scanning the robot needs to be roughly perpendicular to the tissue compared to the insertion plan where the robot needs to strictly follow a straight line.

Phase 2 of the planning goes through all feasible configurations generated in phase 1, until it finds a path to the target. The output of this phase is the generated plan, Π^R, defined as a set of configurations that guides the robot to the target, starting from a selected entry point $\boldsymbol{x}^R_{enter}$. Of note, the entry point and the generated path are defined in the reference frame of the CTR, *i.e*, robot's base, denoted by ^{R}. However, it is assumed that the coordinates of the desired entry point and path are pre-defined from medical images in a global frame of reference as $\boldsymbol{x}^G_t$ and Π^G. We can perform a rigid registration to transform the robot's base to a new

position to register the estimated entry point and the generated path to the global frame. For this purpose, we can use classic point cloud registration algorithms, such as Iterative Closest Point algorithm [95]. The algorithm accepts Π^R and Π^G as inputs to estimate the homogenous transformation matrix that aligns the two cloud points T_C^G. This process is shown in Figure 5.10

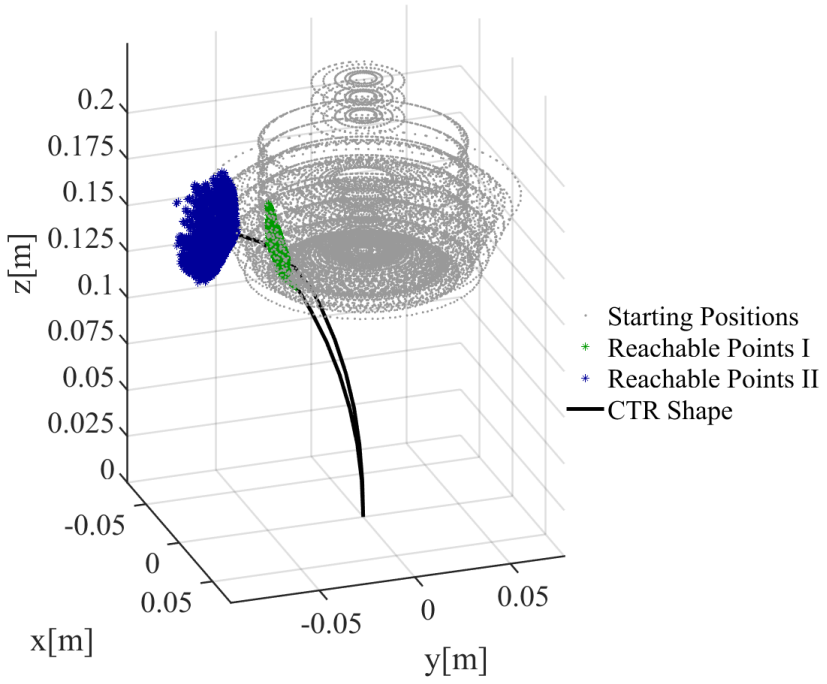


Figure 5.9: Two phases of motion planning. The first phase shows all the tip positions where the movement in the tip z-direction is dominant. The second phase shows the reachable tip positions for the two motion planning problems for selected entry points.

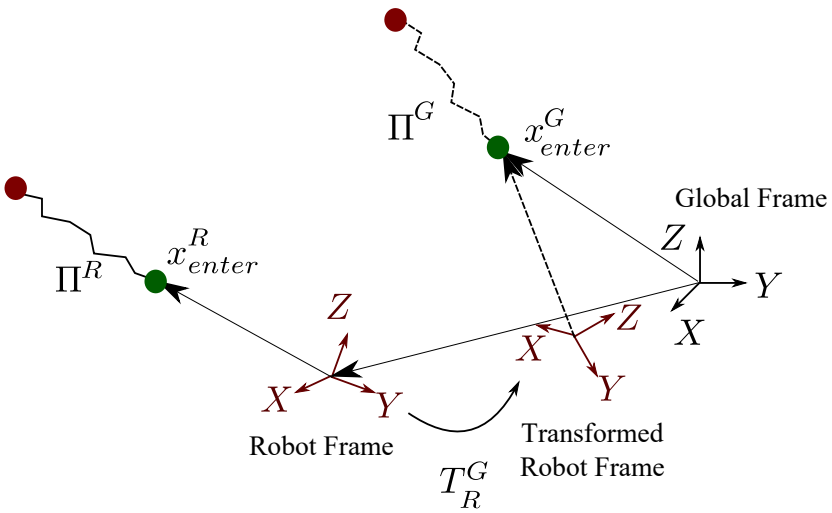


Figure 5.10: Transforming robot coordinate frame to ensure the generated plan and entry point falls on the estimated target from pre-operative images.

5.3.3 Phase 3: Generate a feasible trajectory from the robot's initial configuration to the selected entry configuration

The third phase of the motion planning aims to find an optimal trajectory that brings the robot from any initial configuration to the transformed entry point. We use a similar RRT* algorithm to phase 2. The algorithm is described in Algorithm 1.

The main difference between the algorithms in phase 2 and 3 is that the constrained motion affordance conditions are relaxed in phase 3 as the robot will move in free space. Results of the 3^{rd} phase of the motion planner are shown in Figure 5.11 and Figure 5.12. In the simulations, two random points inside the robot workspace were given as the initial robot configuration and desired entry point.

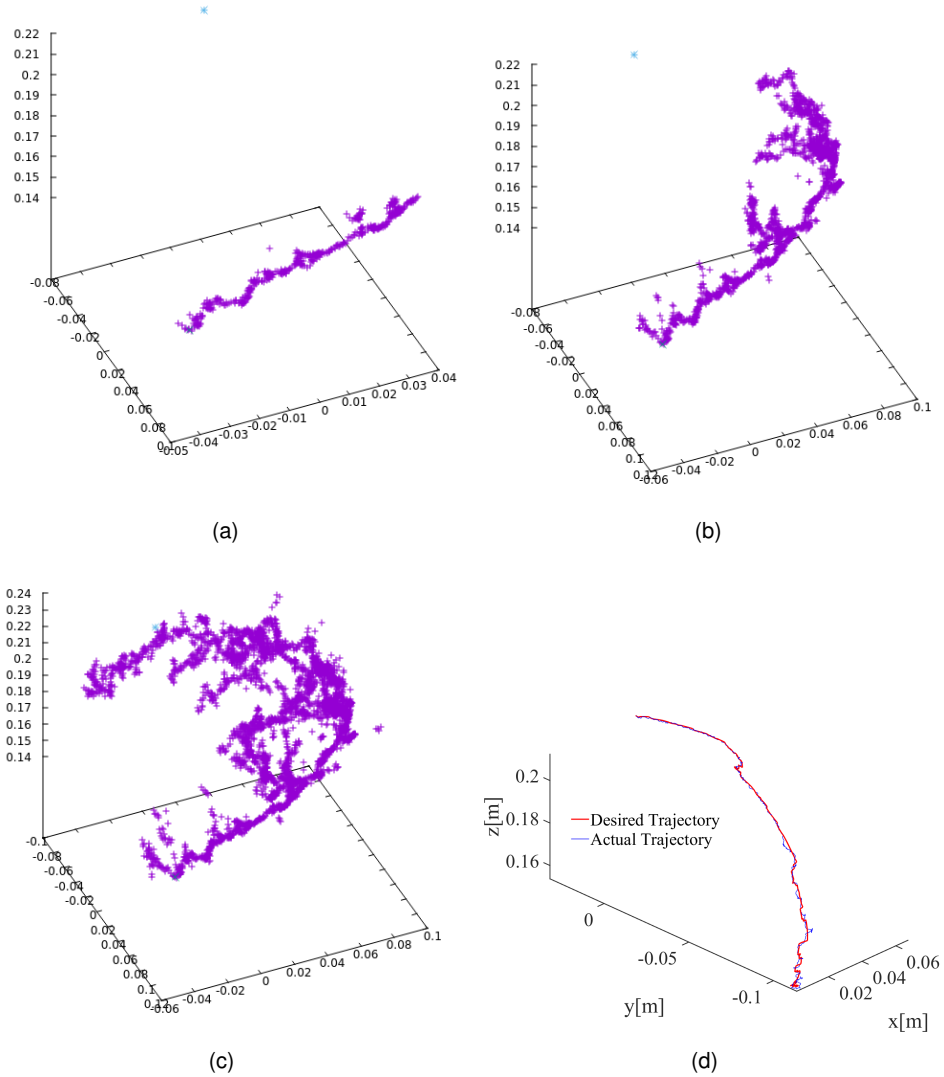


Figure 5.11: Valid configurations starting from an initial position after a) 500, b) 2000, c) 4000 iterations and d) the selected path from the initial position to the final position through consecutive, valid configurations.

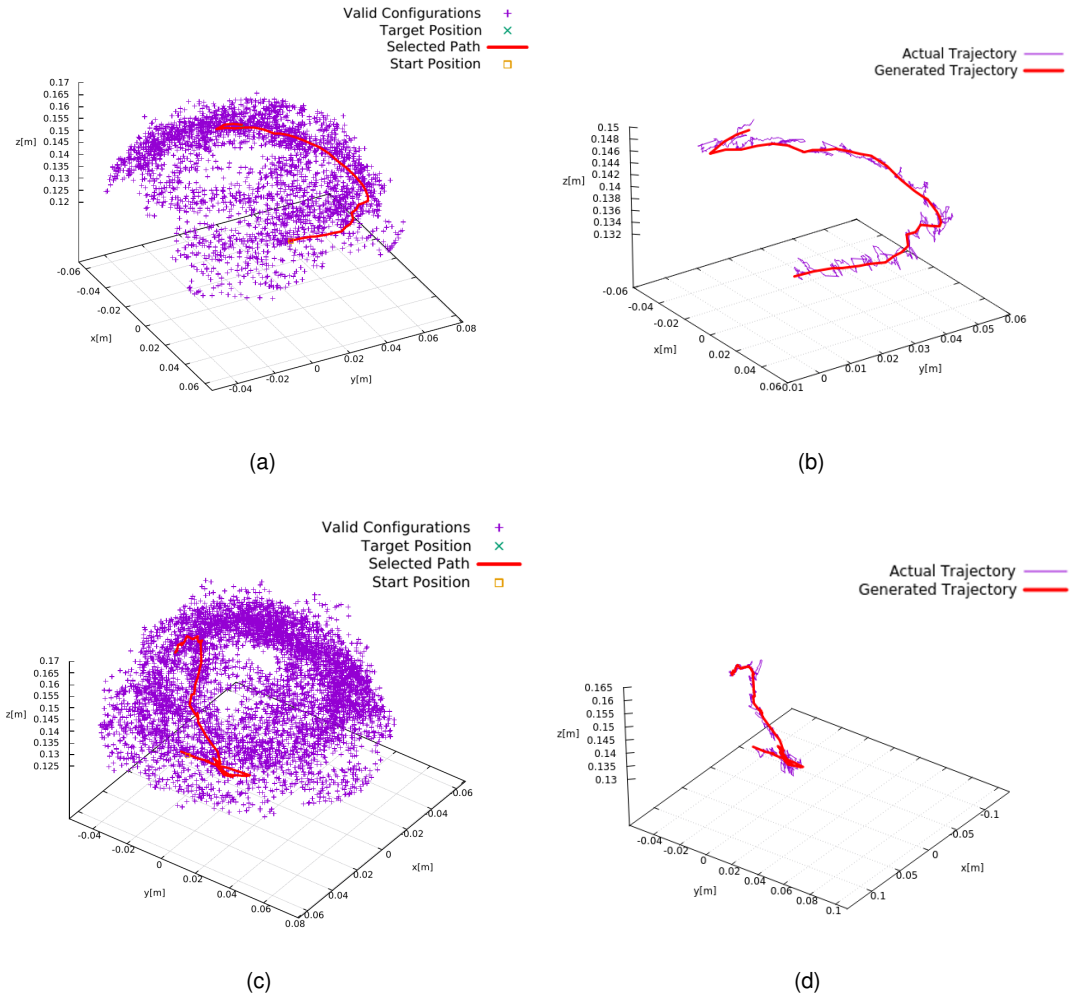


Figure 5.12: The results of the RRT* algorithm are presented. a) and c) show the generated configurations of Γ and the selected path Π, while b) and d) show the desired trajectory Π and actual trajectory with the hybrid controller described in Chapter 4.

Algorithm 1 RRT* algorithm for motion planning phase 3

```
while ||q_new − q_goal|| < δ do
    GenerateNewConfiguration()
    FindNearestNode()
    Contraction()
    FindNewParent()
    if IsObstacleFree then
        if IsStable then
            AddNewNode()
            Rewire()
        else
            AddConfigurationToObstacles()
    else
        DiscardConfiguration()
```

5.4 Experimental Results

To verify the performance of the proposed motion planner, a CTR with two tubes was used with a 5-DOF Electromagnetic tracker (Aurora, NDI) attached to its tip. The mechanical parameters of the CTR are detailed in Chapter 3 (Table 3.3). The Hybrid controller proposed in Chapter 4 was applied to steer the robot's tip along pre-defined trajectories.

The motion planner and the controller were verified through the following three cases:

- (S_1) *Robot following pre-defined trajectories in free-space:* The robot's tip is required to follow a trajectory generated with the RRT* algorithm (phase 3) across its workspace.
- (S_2) *Robot applied as a Steerable Needle:* The robot is required to follow a straight trajectory in its local z-direction. Using the first two phases of the proposed motion planner a straight trajectory was generated with a desired initial configuration.
- (S_3) *Needle-Based Optical Endomicroscopy:* An optical fiber was attached to the CTR's tip providing feedback for the Versiclolour platform. The robot's tip was required to reach multiple points on a phantom tissue along a grid to provide an accurate image to the Versicolour. An offset of 25 mm was applied to the grid in the z-direction from the tip's position. The space between grid points was set to 10 mm. It was necessary to reach each point on the grid from a perpendicular direction to its plane. To achieve these goals, the first two phases of the proposed motion planner were used.

Based on the results of the experiments, Table 5.1 summarizes the accuracy of the tip position during deployment. Figure 5.13 illustrates representative results for the first scenario (S_1) over five trials. Based on the results, the robot's tip is able to follow pre-defined trajectories across the available workspace with a maximum mean error of 2.1 mm.

The results of the second scenario (S_2) show that the CTR can be used as a steerable needle following a pre-generated straight trajectory accurately while it is in constant contact with soft tissue (Fig. 5.18). The experimental setup is shown in Fig. 5.14. It is important to note that the initial configuration and trajectory are chosen using the first and second phases of the proposed motion planner.

Table 5.1: Comparison of Error between desired and actual trajectories. e_{mean} is the average error, σ_e is the standard deviation of the error, e_{max} is the maximum error, and RMSE is the root mean square error. The values are all in millimeters.

	e_{mean}	σ_e	e_{max}	RMSE
S_1**: Path**	2.071	0.950	5.179	0.796
S_2**: Line in Tissue**	0.496	0.704	3.360	0.861
S_3**: Grid Scan**	0.983	0.433	1.849	1.075

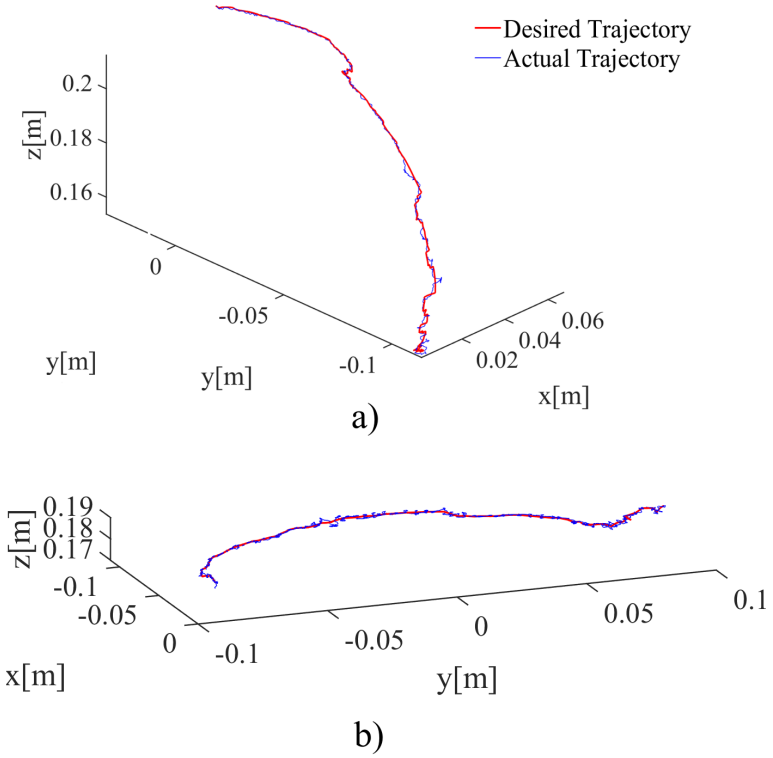


Figure 5.13: Experimental results for the third scenario (S_3): Robot is required to follow pre-generated trajectories across robot's workspace: a) case I, b) case II.

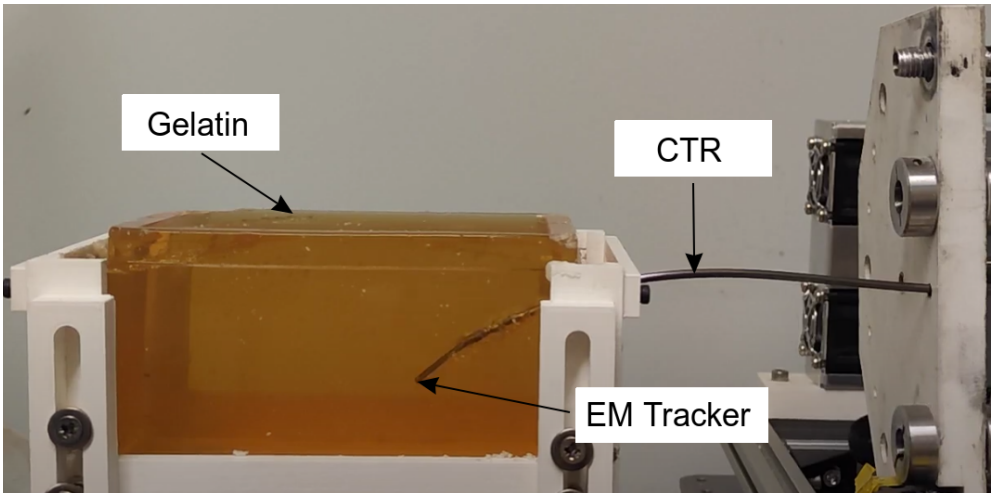


Figure 5.14: Experimental setup for the second scenario (S_2). The tip of the robot is required to follow a straight trajectory inside gelatin.

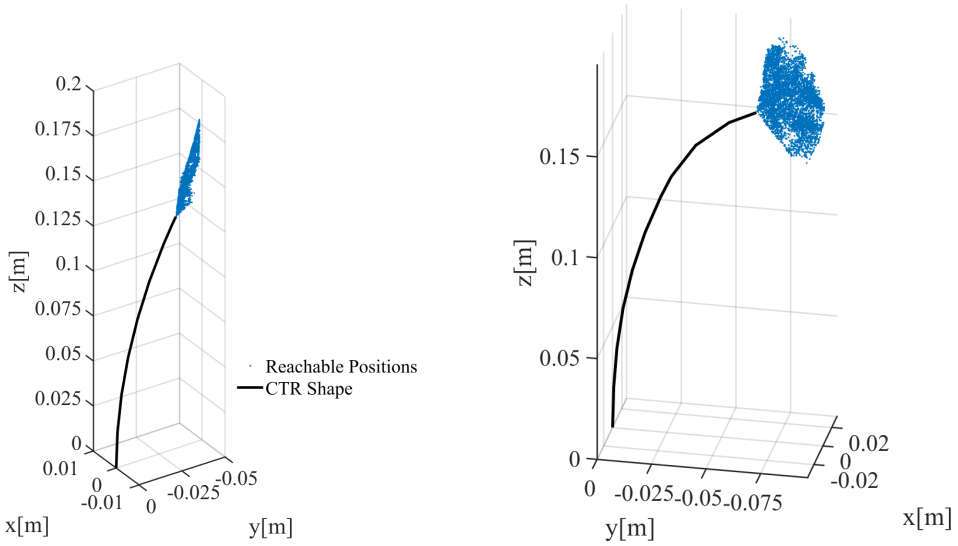
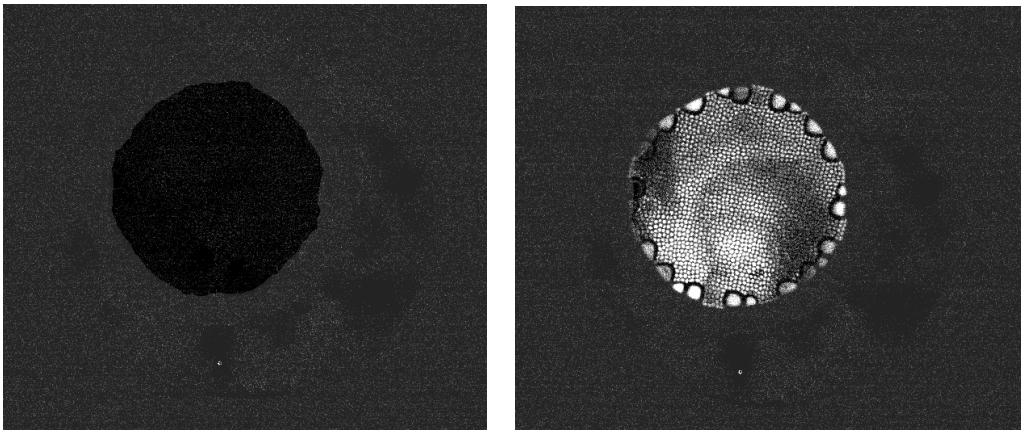


Figure 5.15: Simulation results using the first two phases of the proposed motion planner. (a) making straight movement in the local z-direction (b) reaching multiple points along a grid.

As a final demonstration, we present the third scenario (S_3), which illustrates the proposed CTR's capability to perform a grid scan during an endomicroscopic procedure (Fig. 5.17). During this experiment, a fiber was attached to the tip of the CTR and connected to the Versicolor providing feedback on the procedure. As Fig. 5.19 shows that the tip of the CTR was able to follow the grid trajectory accurately. Figure 5.16(a) shows a typical image of the Versicolour when the tip of the robot is in contact with gelatin, or it does not make any contact and Fig. 5.16(b) shows when it makes contact with one of the stickers. The mean error during the procedure was 0.983 mm. The initial configuration of the robot and the desired trajectory were generated by the first two phases of the motion planner. Of note, the phantom tissue was manually placed in the second (S_2) and third (S_3) experiments. Therefore, the third phase of the motion planner has not been used.

These experimental results verify the proposed motion planner and provide evidence for the potential use of CTRs in optical endomicroscopy and steerable needle applications.



(a) (b)

Figure 5.16: (a) Normal image of the Versicolour, when the tip of the robot is in contact with gelatin or it does not make any contact(b) The image of the Versicolour, when the tip of the robot makes contact with the sticker. The Versicolour is set to show a completely dark image if there is no contact with the target and a bright image when the target is found. The properties of the Versicolour can be altered depending on the requirements.

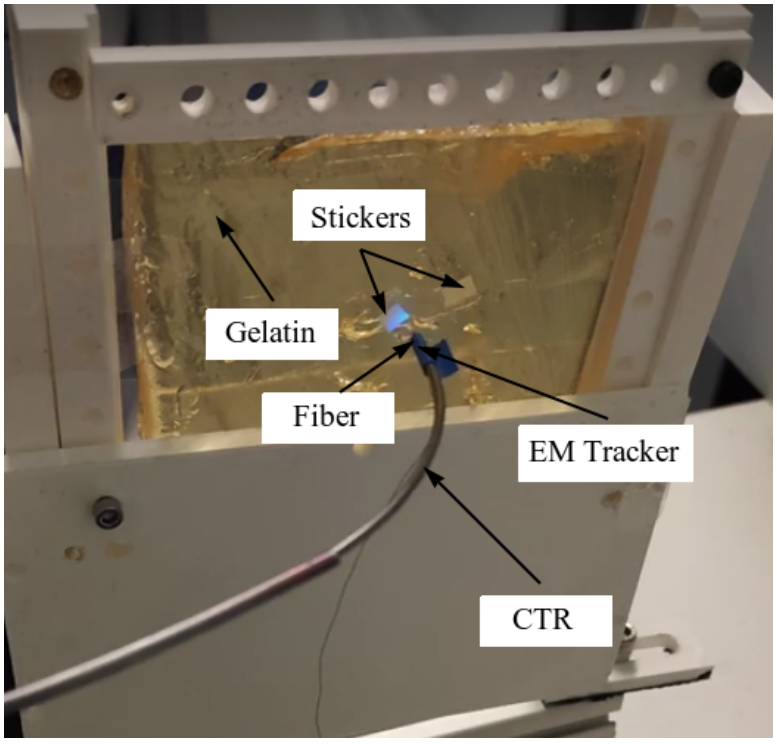


Figure 5.17: Experimental setup for the third scenario (S_3). A fiber is attached to the tip of the robot to provide an image for the Versicolour. The robot is required to reach multiple points along a grid. The goal of this experiment is to detect the stickers on the gelatin.

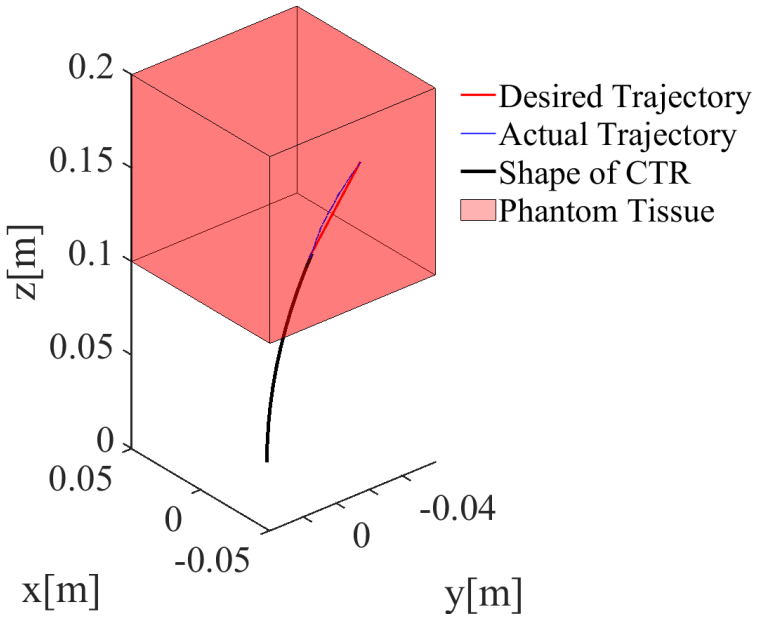


Figure 5.18: Experimental results of the second experiment (S_2). The robot is used as a steerable needle, cutting through gelatin along a straight trajectory.

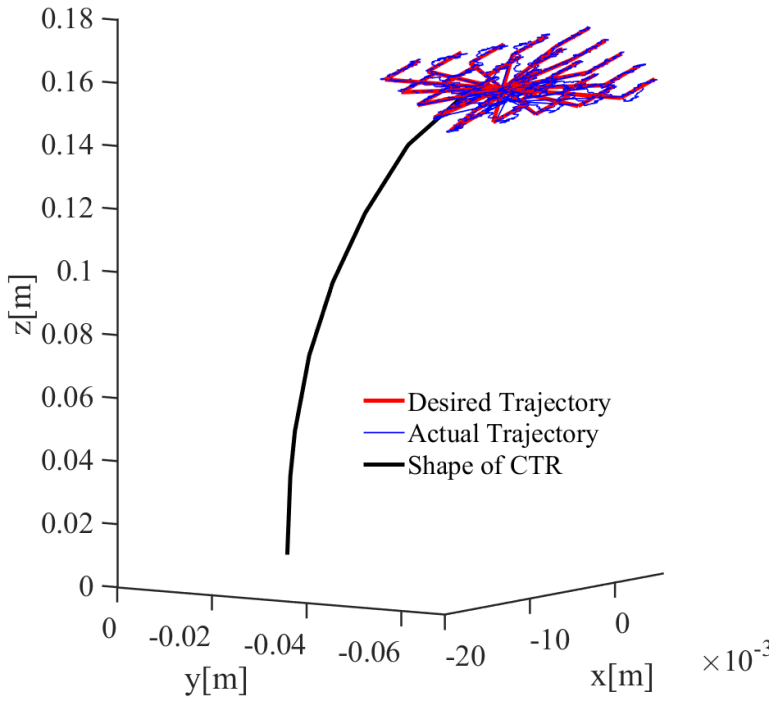


Figure 5.19: Results from the third experiment (S_3). The robot is used for needle-based optical endomicroscopy, reaching multiple points on a gelatin along a pre-defined grid, while a fiber is attached to its tip providing feedback to the Versicolour platform.

5.5 Concluding Remarks

Over the last decade, endomicroscopy has shown to be a promising tool for tissue characterization compared to traditional biopsy. Instead of taking a number of biopsies and examining these with histology, an optical probe can be inserted into the patient and used to obtain high-resolution images of the tissue in real-time. Despite the increased applications and growing research interests in this area, the clinical application of endomicroscopy, however, is limited to surface/near-surface imaging as reliable steerable technologies that can guide the probe inside the tissue are not available. This study aims to overcome this, by proposing a solution for the application of CTRs for needle-based optical endomicroscopy and as steerable needles. In order to accomplish this, a fiber-based multicolour endomicroscopy platform called Versicolour was combined with a CTR. Then, a three-phase motion planner was developed to generate a motion plan for finding the ideal initial configuration for the robot for a given task and to generate a desired trajectory to reach the target. The following factors were important during the motion planning phase: moving perpendicular to the tissue, cutting with minimal force, and avoiding singularities. Moreover, to achieve the required precision for real-life scenarios, the previously proposed hybrid controller was applied. Finally, the proposed motion planner was experimentally verified with a phantom tissue. Based on the experimental results, we demonstrated that our proposed robotic platform is capable of achieving the following two tasks. First, scanning a phantom tissue region, while searching for anomalies in fluorescent images, simulating percutaneous lung biopsy. Second, accurately reaching a target deep inside a phantom tissue at a relative depth of 50 mm. This enables performing a biopsy or targeting cancerous lesions with needle-based ablation procedures to burn or freeze malignant cells.

Chapter 6

Extended Dynamic Mode Decomposition (EDMD)

In Chapter 4, a hybrid approach was discussed for controlling CTRs across robot workspace. Although the proposed controller achieves good accuracy, it still requires the kinematic model as an input, which may not always be available. Therefore, in this chapter, the implementation of a purely data-driven controller is explored to achieve the same goals. It is advantageous to use a purely data-driven approach since it does not rely on the kinematic model, and it is less sensitive to inaccurately identified parameters in the model and controller. The proposed method is achieved by lifting the nonlinear system dynamics into a higher dimensional space where its evolution is approximately linear. This relationship is described by the Koopman operator, which is approximated by applying the Extended Dynamic Mode Decomposition (EDMD). Consequently, this method uses joint positions as inputs and measured tip positions as states, which are then lifted into a higher dimensional space.

The remainder of this chapter is organised as follows. Sections 6.1 presents the contribution of this work. Then, Sections 6.2 details the proposed data-driven controller. In Sections 6.3, an extensive simulation study is performed to tune the controller parameters. Experimental evaluation of the controller and discussion of the results are presented in Section 6.4. Finally, concluding remarks appear in Section 6.5.

The data-driven control method presented in this chapter has also been published [96].

While a video presentation of the experiments is available here:

`https://doi.org/10.1109/LRA.2022.3231490/mm1`

6.1 Contribution

In this chapter, we aim to control the motion of a CTR with unknown dynamics in contact with an unknown environment as shown in Fig. 6.1. The goal is to follow a desired trajectory without having prior knowledge of the external forces acting on the robot or obstacles restricting the robot's motion. It is assumed that we can only measure the Cartesian coordinates of the robot's tip position using commercially available electromagnetic trackers (EMT). To this end, we propose a data-driven method that overcomes the difficulties of the model-based approaches, including predicting the effects of unknown external forces, the robot's dynamics, and unexpected disturbances that might happen in a real setting during the robot's motion. Additionally, the proposed controller requires only 250 samples to effectively learn the robot dynamics, which is significantly less than previous learning-based algorithms. We study the performance of the controller in extensive simulations and experiments. We consider a variety of scenarios including the robot under external forces, the robot in contact with an obstacle, and the robot cutting through phantom tissue, simulating percutaneous needle-based interventions. Our algorithm is available online.[1]

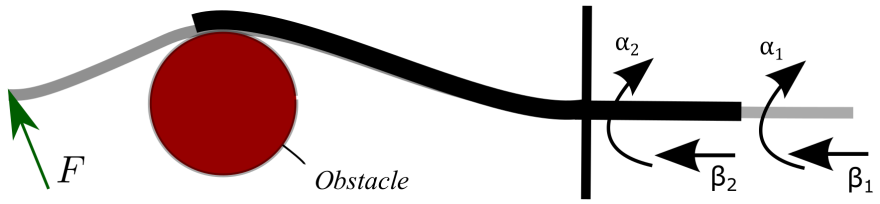


Figure 6.1: Illustration of a CTR with two tubes in contact with an obstacle and under external forces. The actuation variables α_i denotes the proximal base rotation of the i^{th} tube, while β_i is the translation joint input of the i^{th} tube.

6.2 Methodology

Here, we present a holistic approach to model the dynamics of a closed-loop controlled CTR. We use several samples from the system to develop a linear state space model of the CTR, *i.e.*, a model that accepts joint inputs and measurements of the robot's tip position over a fixed period of time and predicts the future position of the robot's tip. Each sample at time t_k contains the Cartesian coordinates of the robot's end-effector position $\boldsymbol{x}$ and the robot control inputs $\boldsymbol{q} = [\beta_1 \quad \beta_2 \quad \alpha_1 \quad \alpha_2]$ at time t_k. The proposed model is linear, however, it captures the nonlinear dynamics of the robot and interaction forces. It is not linearizing the system in the sense that it neglects higher-order nonlinear terms. Instead, it is transforming the robot's dynamics into a higher dimensional space, where the mapping between inputs and outputs

1. https://github.com/SIRGLab/CTR-EDMD.git

is linear. This will reduce the system's nonlinearity at the cost of increasing the system's dimensionality. In particular, rather than describing the evolution of a dynamical system's state directly, which may be a nonlinear mapping, the Koopman operator [97] is employed to describe the evolution of continuous scalar-valued functions of the state. Koopman operator has been previously proposed for data-driven modelling of nonlinear systems. A review of Koopman applications in nonlinear control can be found in [98]. Later, we apply a numerical approximation algorithm known as Extended Dynamic Mode Decomposition (EDMD) [99] to discretize the linear system. EDMD is an efficient numerical method previously proposed for reducing the dimensionality of linear systems [100]. The final model is a linear a dynamical system that allows established linear control design methodologies to be used to design controllers for the robot. Finally, we employ a linear model predictive controller that employs the data-driven dynamic model to control the position of the CTR. A block diagram of the proposed control strategy is shown in Fig. 6.2.

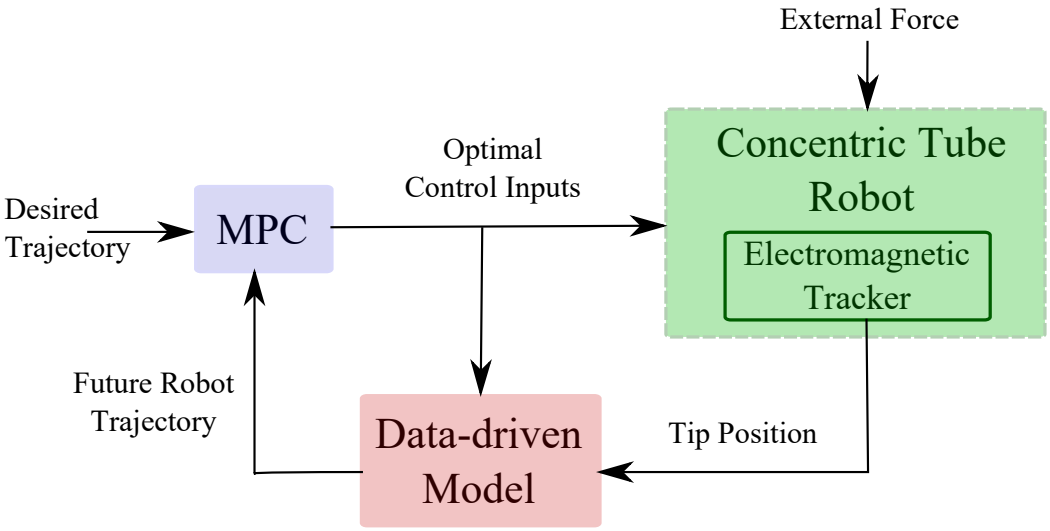


Figure 6.2: A block diagram of the proposed control strategy.

6.2.1 Data-driven Modelling via Dynamic Mode Decomposition

This section describes a data-driven approach for modelling the dynamics of a CTR as a linear state space system. The basic idea is to lift (or embed) the nonlinear dynamics into a higher dimensional space where its evolution is approximately linear. For this purpose, we use a linear operator known as the Koopman operator. Later, we apply the Extended Dynamic Mode Decomposition (EDMD) to compute a finite-dimensional approximation of the operator to form an input/output discrete dynamical system representing the motion of the CTR.

Without loss of generality, we assume the CTR is composed of two tubes. We can summarize the dynamic behaviour of the CTR as

$$\boldsymbol{x}^{+} = f(\boldsymbol{x}, \boldsymbol{q}), \tag{6.1}$$

where $\boldsymbol{x} \in R^3$ is the state of the system representing the Cartesian coordinates of the robot's tip, $\boldsymbol{q} \in R^4$ are the 4 control inputs shown in Fig. 6.1, $\boldsymbol{x}^{+}$ is the successor state and f is the dynamic transition mapping.

The Koopman operator is an infinite-dimensional and linear operator originally developed to describe the evolution of nonlinear autonomous systems [97]. More specifically, it describes the evolution of functions of the states of a nonlinear system. We call these functions of states *observables*. To this end, we define a set of scalar-valued observables as a mapping

$$g(\boldsymbol{x}, \boldsymbol{q}) : \mathbb{R}^3 \times \mathbb{R}^4 \to \mathbb{R}, \tag{6.2}$$

where g belongs to an infinite dimensional Hilbert space $\mathscr{H}$ [101]. Now, using the extension of the Koopman operator it is possible to incorporate control inputs into the analysis. According to this, the Koopman operator with inputs and control (KIC) can be used to capture the control inputs' effects on the system dynamics. It is defined as a linear operator $\mathscr{K} : \mathscr{H} \to \mathscr{H}$ such that

$$\mathscr{K}\, g(\boldsymbol{x}, \boldsymbol{q}) \equiv g(f(\boldsymbol{x}, \boldsymbol{q}), \boldsymbol{q}^{+}). \tag{6.3}$$

Linear operators from Hilbert space to Hilbert space have associated eigenvalues and eigenfunctions, *i.e.*,

$$\mathscr{K}\, \phi_i(\boldsymbol{x}, \boldsymbol{q}) = \lambda_i \phi_i(\boldsymbol{x}, \boldsymbol{q}), \tag{6.4}$$

where ϕ_i is the i^{th} eigenfunction and λ_i is the i^{th} eigenvalue. Consequently, the observable functions can be written as a linear combination of all the eigenfunctions (since they form a basis)

$$g(\boldsymbol{x}, \boldsymbol{q}) = \sum_{i=1}^{\infty} \phi_i(\boldsymbol{x}, \boldsymbol{q}) v_i \tag{6.5}$$

where v_i is called the i^{th} Koopman mode associated with the i^{th} Koopman eigenfunction. In what follows, we approximate the infinite-dimensional KIC system with a finite-dimensional system of size N:

$$\mathscr{K}\, g(\boldsymbol{x}, \boldsymbol{q}) \approx \sum_{i=1}^{N} \lambda_i \phi_i(\boldsymbol{x}, \boldsymbol{q}) v_i = K g(\boldsymbol{x}, \boldsymbol{q}), \tag{6.6}$$

where K denotes the linear map from the observables' space to the finite-dimensional approximation of the KIC's resulting Hilbert space, which we call the *lifted space*.

We compute this approximation to the Koopman operator using a method called extended dynamic mode decomposition [100]. First, the observables are partitioned into functions of the state Y_x, input Y_q, and both $Y_{x,q}$

$$Y = \begin{bmatrix} Y_{\boldsymbol{x}} \\ Y_{\boldsymbol{q}} \\ Y_{\boldsymbol{x},\boldsymbol{q}} \end{bmatrix} = \begin{bmatrix} g_1(\boldsymbol{x},\mathbf{0}) \\ \vdots \\ g_l(\boldsymbol{x},\mathbf{0}) \\ g_{l+1}(\mathbf{0},\boldsymbol{q}) \\ \vdots \\ g_{l+p}(\mathbf{0},\boldsymbol{q}) \\ g_{l+p+1}(\boldsymbol{x},\boldsymbol{q}) \\ \vdots \\ g_{l+p+j}(\boldsymbol{x},\boldsymbol{q}) \end{bmatrix} = \mathbf{g}(\boldsymbol{x},\boldsymbol{q}), \tag{6.7}$$

where l, p, and j denote the number of basis functions used to estimate the states, inputs, and input-output coupled dynamics, respectively. Applying the Koopman operator to the observables gives

$$Z = \begin{bmatrix} Z_x \\ Z_q \\ Z_{x,q} \end{bmatrix} = \mathbf{g}(f(\boldsymbol{x},\boldsymbol{q}),\boldsymbol{q}^+) \tag{6.8}$$

We want the EDMD approximation of KIC to be linear and affine. Therefore, we set $Y_{x,q} = 0$. Keeping $Y_{x,q}$ would result (in the simplest case) in a bilinear approximation to the dynamical system. Predictors with bilinear form are not immediately suited for control design. Therefore, we decide to set $Y_{x,q} = 0$. This potentially reduces the model's accuracy. However, this results in a linear control system that is computationally more efficient. Additionally, we are not concerned with the internal dynamics of the input, thus $Z_{x,q} = Z_q = 0$. We note that by definition then $Z_x = Y_x^+$.

The choice of functions in Y_x, Y_q and Z_x are a design choice. A common choice is polynomial functions of varying order [102]. In this work, we set the state vector observables to be

$$Y_x = \begin{bmatrix} x_1 & x_2 & x_3 & x_1^2 & x_2^2 & x_3^2 & x_1^2x_2 & x_1^2x_3 & x_2^2x_1 & x_2^2x_3 & x_3^2x_1 & x_3^2x_2 \end{bmatrix}^T \tag{6.9}$$

and the input vector observables to be

$$Y_q = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 \end{bmatrix}^T = \begin{bmatrix} \beta_1 & \beta_2 & \alpha_1 & \alpha_2 \end{bmatrix}^T. \tag{6.10}$$

Assuming that we collect a dataset of tip positions and control inputs at m consecutive time samples, we get

$$\begin{bmatrix} Z^0 & Z^1 & \dots & Z^m \end{bmatrix} = \begin{bmatrix} A & B \end{bmatrix} \begin{bmatrix} Y_x^0 & Y_x^1 & \dots & Y_x^m \\ Y_q^0 & Y_q^1 & \dots & Y_q^m \end{bmatrix} \tag{6.11}$$

where $K = \begin{bmatrix} A & B \end{bmatrix}$, superscripts $0, \cdots, m$ denote the time steps, with m commonly known as the DMD horizon showing the number of samples used to estimate the linear model. The elements of $Z_{x,q}$ are ignored here since we have set those to zero (as well as $Y_{x,q}$). We note that functions of states in Y_x, Y_q, and Z_x are computed directly from the tip measurements and control inputs. Now we can use the EDMD algorithm to estimate A and B. EDMD aims to calculate A and B so that we arrive at the linear differential equation

$$Y_x^+ = AY_x + BY_q. \tag{6.12}$$

Given the Datasets $Z^0, Z^1, ..., Z^m$, $Y_x^0, Y_x^1, ..., Y_x^m$, and $Y_q^0, Y_q^1, ..., Y_q^m$, first we Construct an input matrix

$$Y = \begin{bmatrix} Y_x & Y_q \end{bmatrix} = \begin{bmatrix} Y_x^0 & Y_x^1 & \dots & Y_x^m \\ Y_q^0 & Y_q^1 & \dots & Y_q^m \end{bmatrix}, \tag{6.13}$$

and an output matrix

$$Z = \begin{bmatrix} Z^0 & Z^1 & \dots & Z^m \end{bmatrix}. \tag{6.14}$$

Next, we compute the singular value decomposition (SVD) of the input matrix and partition the result into the first l rows associated with Y_x and the next p rows associated with Y_q.

$$Y = U\Sigma V^* = \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} \Sigma V^*. \tag{6.15}$$

Where * is the conjugate transpose of a matrix. Of note, * is overloaded and here it does not have the same meaning as in the model (3.1), where it denotes the initial curvature. Similarly, we compute the SVD of the output matrix

$$Z = \hat{U}\hat{\Sigma}\hat{V}^*. \tag{6.16}$$

Finally, we can estimate matrices A and B as

$$\begin{aligned} A &= \hat{U}^* Z V \Sigma^{-1} U_1^* \hat{U} \\ B &= \hat{U}^* Z V \Sigma^{-1} U_2^*. \end{aligned} \tag{6.17}$$

Fig. 6.3 summarizes the proposed modelling approach. The model given in (6.12) will accept the m samples of input $\boldsymbol{q}$ and output $\boldsymbol{x}$ passed through the observables as Y_q and Y_x and predicts the successor observable states Y_x^+. We note that, based on (6.9), the first three elements of Y_x^+ will be the predicted future output of the plant, *i.e.*, $\boldsymbol{x}^{m+1}$. Additionally, the rank of A describes the dimension of the lifted space (N in (6.6)) and is equal to the number of state vector observables. Based on (6.9), N is equal to 12. One can always select more observables to increase the model accuracy at the cost of reducing the EDMD's computational efficiency. In Section 6.3, simulations are performed to demonstrate that 12 observables are sufficient for estimating the lifted space of the CTR.

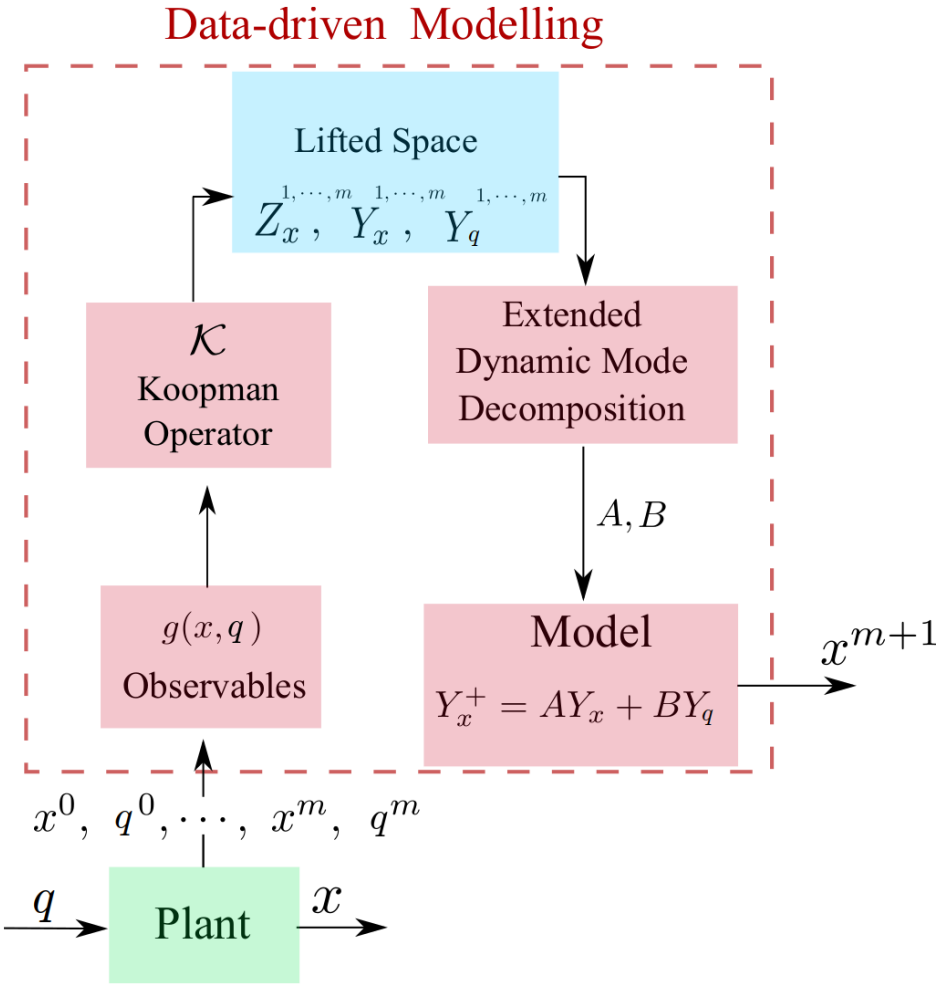


Figure 6.3: A block diagram of the proposed data-driven modelling approach.

6.2.2 Model Predictive Control

Here, we employ a model predictive control (MPC) algorithm that uses the linear data-driven model of CTRs in (6.12) to steer the CTR over a pre-defined desired trajectory $\boldsymbol{x}_d$. The model predictive controller aims to find the control inputs ($\boldsymbol{q}$) to drive the states ($\boldsymbol{x}$) to some reference value over a fixed horizon by solving the following quadratic program:

$$\begin{aligned}
&\text{minimize: } \sum_{i=1}^{M} (\boldsymbol{x}^i - {\boldsymbol{x}_d}^i)^T \mathbf{Q} (\boldsymbol{x}^i - {\boldsymbol{x}_d}^i) + {\boldsymbol{q}^i}^T \mathbf{R} \boldsymbol{q}^i \\
&\text{with respect to: } \boldsymbol{q} \\
&\text{subject to:} \\
&Y_x^+ = \mathbf{A} Y_x + \mathbf{B} Y_q \\
&\boldsymbol{q}_L < \boldsymbol{q}^i < \boldsymbol{q}_U .
\end{aligned} \tag{6.18}$$

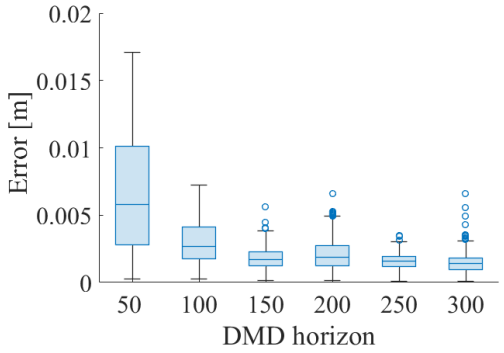
where $i = 1, \cdots, M$ denotes the time instant, M is the MPC horizon, $\mathbf{Q}$ and $\mathbf{R}$ are positive definite matrices penalizing tracking error and the control inputs. $\boldsymbol{q}_L$ and $\boldsymbol{q}_U$ denote the upper and lower limits of the joint inputs. In this work, we used the algorithm in [103] to solve the MPC problem. The algorithm employs the Nesterov Accelerated Gradient method to rapidly estimate the gradient of the cost function and improve the computational efficiency of the MPC. The gradient is later used in a gradient descent algorithm for updating control variables while minimising the cost function.

6.3 Simulation Study

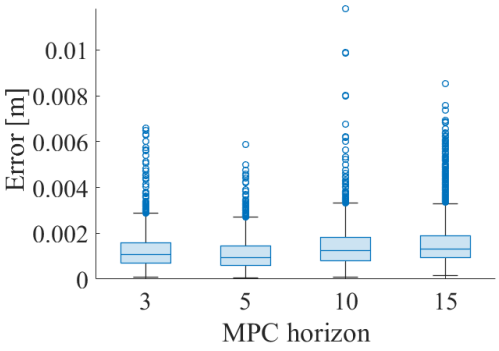
In this section, we perform simulations to evaluate the performance of the proposed controller. Simulation environment for the CTR is developed using the mathematical model of the CTR presented in [60]. The robot is composed of two tubes. Mechanical characteristics of the robot used in the simulations are the same as detailed in Section 3.5.1. These are the parameters of an actual robot used in the experiments. The robot's tip was steered to follow a square trajectory with 20 mm base at a velocity of 1 mm/s.

In the first set of simulations of this square trajectory, we evaluated the effect of DMD horizon (m) on the accuracy of the controller. Six different horizons ranging from 50 to 300 were selected. To simulate the effects of noise in the sensory feedback, we added random Gaussian noise with a standard deviation of 2 mm to the measured tip position. Results are summarized in Fig. 6.4(a). Furthermore, we simulated a scenario where the robot's tip's Cartesian coordinates are compared with only the first three states of the Lifted Space, which based on (6.9) correspond to the robot tip coordinates. The robot is moved randomly and the DMD horizon was set to 250. After the 250 samples, the robot tip position is compared with the

first three states of the lifted space. We note that this is an open-loop simulation. The average error measured over ten trials is 0.3 mm with a standard deviation of 0.1 mm. Based on these results, we selected the DMD horizon m to be equal to 250 to be later used in the experiments, as it was found to achieve the minimum error.



(a)



(b)

Figure 6.4: (a) Error bars comparing CTR's tip position error with respect to the DMD Horizon. Starting from the middle of the box, the line inside the box corresponds to the median error, box ends indicate the 25th and 75th percentiles, and the ends of the dashed lines are the maximum and minimum errors. (b) Error bars comparing CTR's tip position error with respect to the MPC Horizon.

In the next simulation, we evaluated the effects of MPC Horizon (M). Four different horizons ranging from 3 to 15 sample times were selected. The result of trajectory tracking error for various MPC horizons are shown in Fig. 6.4(b). The results did not show any significant difference between the different horizons. We selected the largest horizon (*i.e.*, 15). The proposed data-driven approach accepts the system states, *i.e*, robot tip position, and corresponding control inputs for a fixed period of time (DMD horizon). Next, it develops a linear map between control inputs and robot states in a higher dimensional space called the lifted space. As shown in Fig. 6.4(a), an optimal number of 250 samples are sufficient to learn the robot behaviour moving at a velocity of 1 mm/sec. We note that this is a continuous learning

process. Therefore, a change in external disturbance will affect the robot states and the EDMD has the ability to adapt quickly and update the model accordingly. However, if the robot is moving too fast, the sampling frequency must be increased so that the model can capture the rapidly varying dynamics of the system.

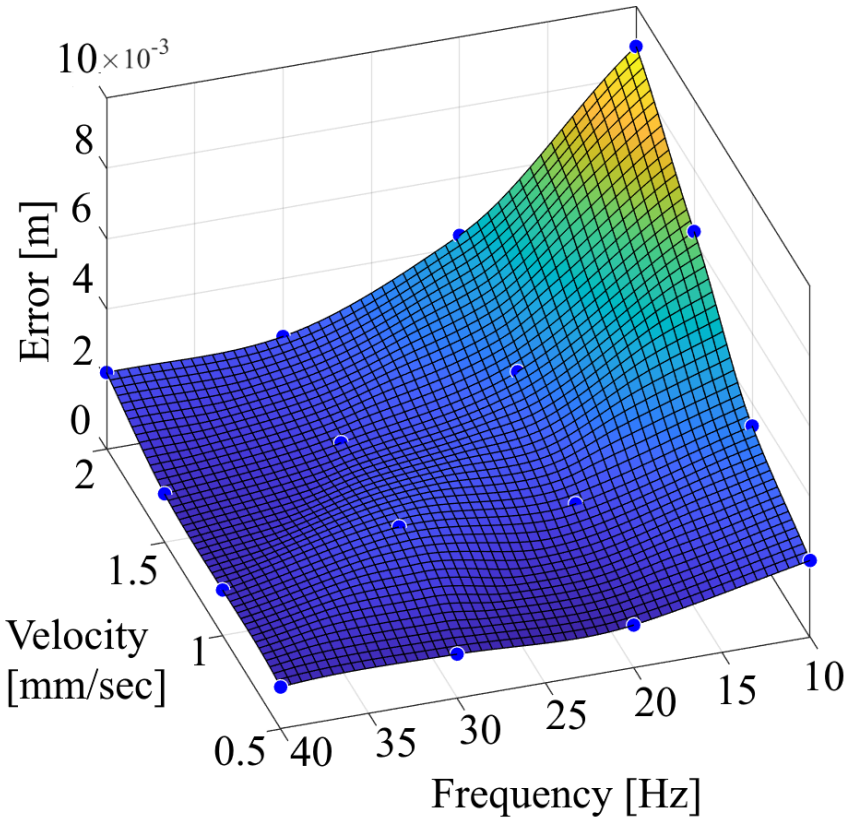


Figure 6.5: Tip position error with respect to sampling frequency and desired velocity. The blue dots on the plot are the results from the simulations, the surface on the plot was fitted on these points.

To investigate the effect of sampling frequency on the proposed model and controller, we performed 16 trials; in each trial, the robot was asked to follow the same trajectory at velocities of 0.5, 1, 1.5, and 2 mm/sec, with sampling frequencies varying from 10 to 40 Hz. We note that the maximum sampling frequency of the electromagnetic tracking sensor used in the experiment is 40 Hz. Therefore, we limit the sensor frequency in the simulations to 40 Hz. Results are summarized in Fig. 6.5. The simulation demonstrates that the tracking error generally increases with respect to sampling frequency and tip velocity. Moreover, it is evident that the tracking error at higher velocities can be minimized by selecting a higher sampling frequency. Based on this analysis, we selected 1 mm/s as the robot's desired tip velocity and 40 Hz as the sampling frequency in our experiments.

6.3.1 EDMD Generalization to a CTRs with 3 Tubes

To test the performance of the proposed algorithm on CTRs with 3 tubes, we carried out two simulations. The parameters of the 3-tube CTR used in the simulations are detailed in Chapter 3 (Table 3.3). The newly added tube has a curvature of 21.3, which is based on a real robot discussed in [27].

Extending the proposed method to 3 tubes does not require any major changes in the lifting functions. The only difference is that the number of control inputs for a 3-tube CTR is 6 instead of 4. Control inputs in this case are three translational joint inputs and three rotational joint inputs. Accordingly, we updated the input vector observables in 6.10 as follows

$$Y_q = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 & q_5 & q_6 \end{bmatrix}^T = \begin{bmatrix} \beta_1 & \beta_2 & \beta_3 & \alpha_1 & \alpha_2 & \alpha_3 \end{bmatrix}^T. \tag{6.19}$$

Two simulation studies were carried out. First, the robot's tip was controlled to follow a square trajectory with a base length of 20 mm. Next, the robot's tip was steered to follow a circular trajectory with a 30 mm radius. 2 mm Gaussian noise was added to the tip position to simulate realistic sensory feedback. Results are shown in Fig. 6.6. The root mean squared error is 1.06 mm for the square trajectory and 1.27 mm for the circular trajectory. The results demonstrate that the proposed controller provides similar performance to a CTR with 2 and 3 tubes.

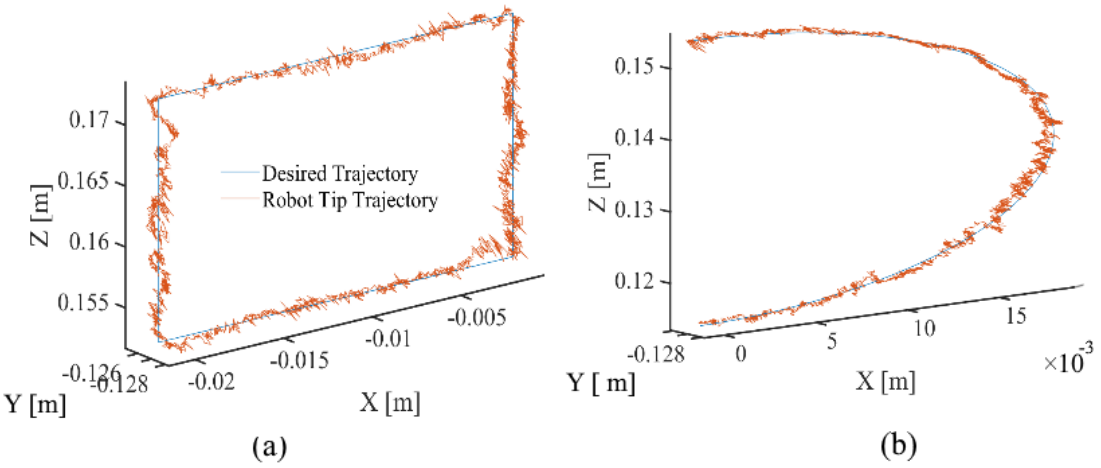


Figure 6.6: Simulation results on a CTR with 3 tubes. (a) Tracking a square trajectory. (b) Tracking a circular trajectory.

6.3.2 Open-loop Control with EDMD

To verify the method and to provide a way for a better understanding of the model, an open-loop control simulation study was carried out. We simulated a scenario where the robot's tip's Cartesian coordinates are compared with only the first three states of the Lifted Space, which are defined in (6.9). Based on this, the first three states correspond to the robot's tip coordinates. However, the proposed method models the robot's behaviour in a lifted space, where the input-output relationship is linear. Then, the robot is controlled in that space. Therefore, the process of control and modelling is coupled. Additionally, the mapping between the robot space and the lifted space is nonlinear. Therefore, comparing the lifted space with robot space is not always feasible. For instance, the lifted space states include terms such as $x_1^2 x_2$ and $x_3^2 x_2$ with $\boldsymbol{x} \in R^3$ defined as Cartesian coordinates of robot tip.

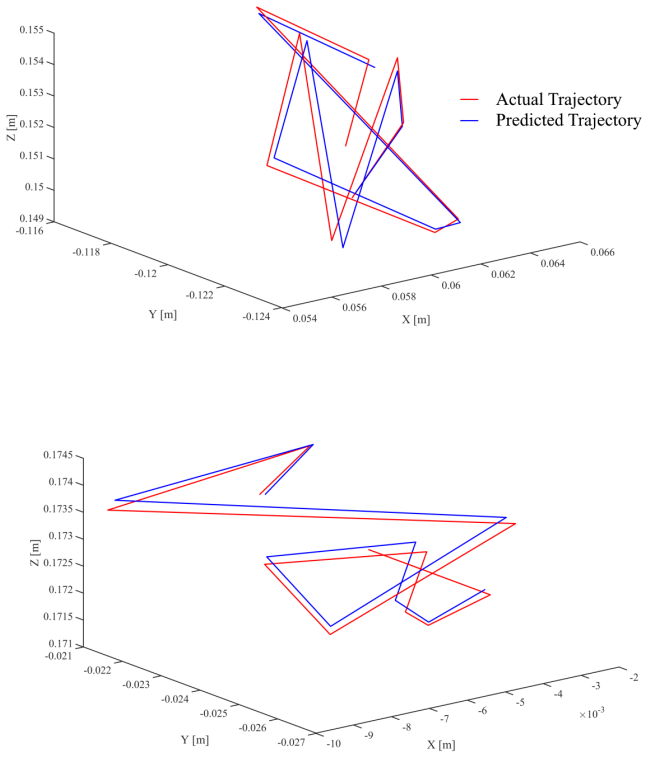
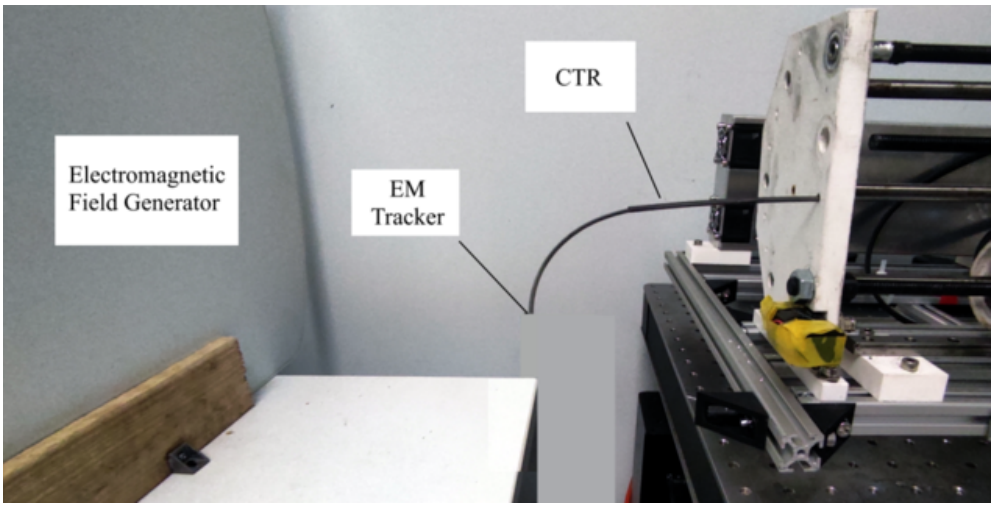


Figure 6.7: A comparison between the robot tip position predicted by the data-driven model and the simulated model. The robot inputs are generated randomly.

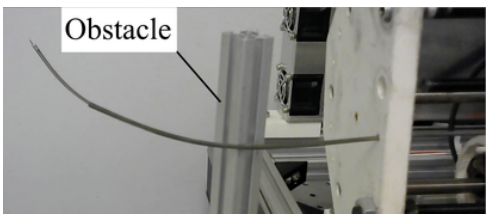
According to this, the robot is moved randomly, while the DMD horizon was set to 250. After the 250 samples, the robot tip position is compared with the first three states of the lifted space. We note that this is open-loop. The average error measured over ten trials was 0.3 mm. Representative results of the simulation are shown in the Fig. 6.7.

6.4 Experiments and Discussions

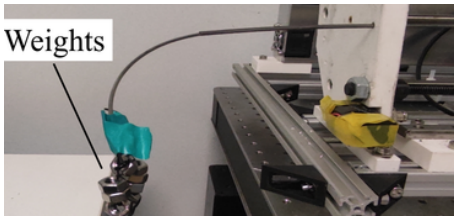
Several experiments were performed to evaluate the performance of the proposed controller. The experimental setup consists of a CTR with two tubes and a 5-DOF electromagnetic tracker (Aurora, NDI) attached to its tip (Fig. 6.8(a)). Based on the manufacturer datasheet, the electromagnetic tracker has a mean accuracy of 0.7 mm and a maximum error of 1.8 mm. The controller was implemented in Robot Operating System (ROS) in C++ and tested on a Desktop Computer with Intel(R) Core(TM) i9-12900K CPU processor and 32.0 GB of Memory.



(a)



(b)



(c)

Figure 6.8: (a) Experimental setup. An electromagnetic tracker is placed at the tip of the robot to measure the robot's tip position. (b) Robot in contact with an obstacle. (c) 20 gram weight is connected to the robot's tip.

The following four scenarios were considered in the experiments:

(S_1) *Robot following a pre-defined trajectory in free space:* The controller was tested on the CTR in free space. The robot's tip was steered to follow three different trajectories: (i) a square trajectory with 20 mm base length, (ii) a circular trajectory with a 30 mm radius, and (iii) a longer trajectory towards the edge of the robot's workspace, which was given as a sequence of equally distanced random points. 10 trials are performed for each trajectory.

(S_2) *Robot under unknown external forces:* The controller was tested while a point load of 20 grams was applied at the tip of the robot (Fig. 6.8(c)). The controller is tasked to follow a square trajectory with 20 mm base length. 10 trials are performed.

(S_3) *Robot in the presence of an obstacle:* The robot's tip was steered to follow a sequence of points while a fixed obstacle was obstructing the robot's motion (Fig. 6.8(b)).

(S_4) *Robot in contact with phantom tissue:* The robot was required to perform multiple tasks while it was in contact with a phantom tissue.

Based on the simulation study presented in Section 6.3, the sampling frequency and the desired tip velocity in all the scenarios were selected as 40 Hz and 1 mm/sec, respectively. In addition to this, we selected the MPC parameters $\mathbf{Q} = 5 \times 10^7 \mathbf{I}$, and $\mathbf{R} = \mathbf{I}$. The MPC horizon was set to 15. The DMD horizon was set to 250 as well. Prior to the experiments, the robot's joint inputs were altered randomly for 10 seconds, while the corresponding tip positions were recorded. This data set was used to initialize the data-driven model.

Representative results of the trajectory tracking in the first scenario (S_1) are shown in Fig. 6.9(a-c). The proposed data-driven algorithm is capable of following various trajectories with high accuracy. The mean tracking error and standard deviation of error are reported in Table 6.1.

In the second scenario (S_2), a weight of 20 grams was attached to the robot's tip. Of note, the 20 gram weight at the tip of the CTR causes a significant change in the CTR's shape and tip position as shown in Fig. 6.8(c). The controller was able to adapt to these changes without any prior knowledge of the weight and accurately follow the desired trajectory (Fig. 6.10). The mean error and the standard deviation of the error over 10 trials are 1.7 mm and 2 mm, respectively.

Table 6.1: Comparison of Error between desired and actual trajectories. The mean error and the standard deviation of error are reported over 10 trials. The values are all in millimeters.

	Scenario S1			Scenario S2
Trajectory	Square	Circular	Point series	Square with weight
Mean error	1.40	0.748	2.40	2.00
std	0.97	0.450	1.70	1.70

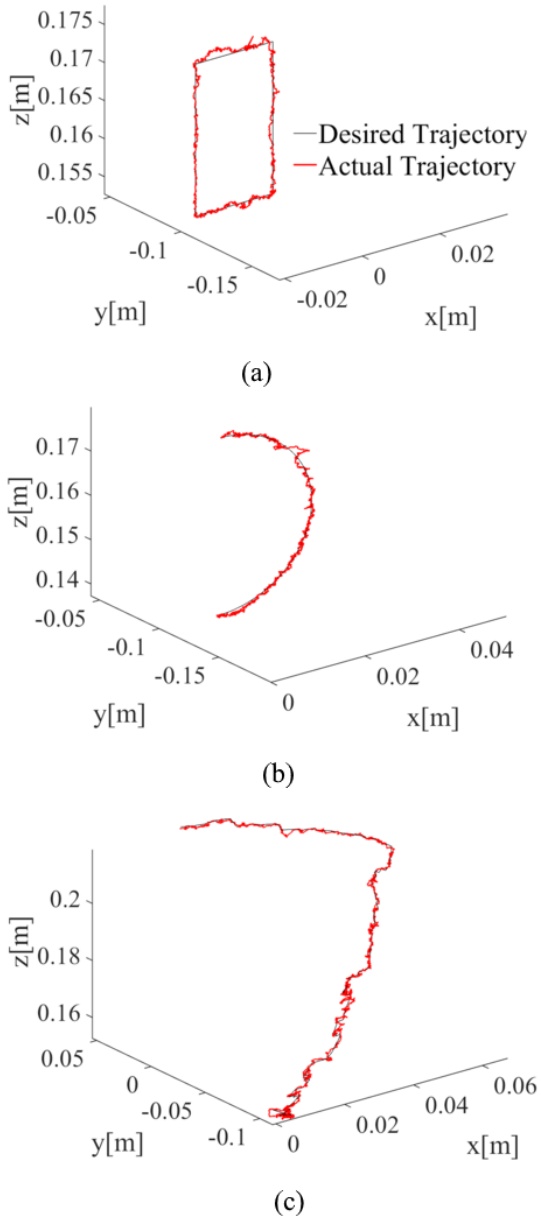


Figure 6.9: Representative experimental results for trajectory tracking in the 1^{st} scenario (S_1). (a) Square Trajectory. (b) Circular Trajectory. (c) A long trajectory towards the edge of the robot's workspace.

In the third scenario (S_3), the robot was tasked to follow several target points while it was in continuous contact with an obstacle. The target positions were selected to be at least 1 cm apart. The controller stops once it reaches the vicinity of the target and the error becomes less than 2 mm. This value was selected based on the accuracy of the electromagnetic sensor (1.8 mm). Results are shown in Fig. 6.11. In all cases, the robot was able to reach the desired target positions with a maximum error of 2 mm.

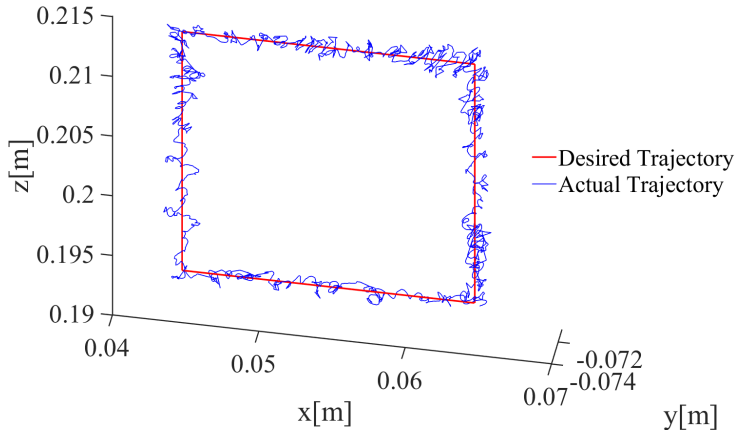
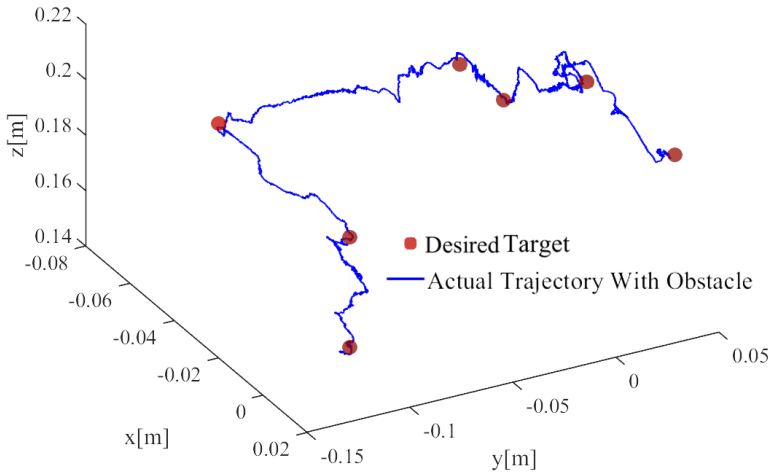
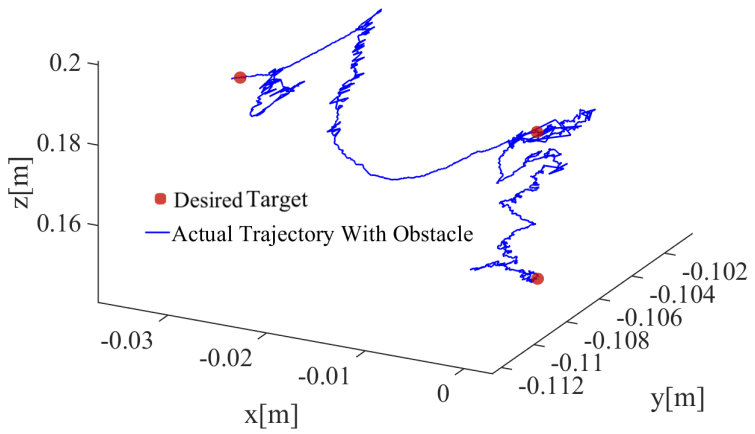


Figure 6.10: Result for the second experimental scenario (S_2), a 20g weight is attached to the robot's tip while it's following a square trajectory. (a) Robot tip trajectory. (b) Tracking error.



(a)

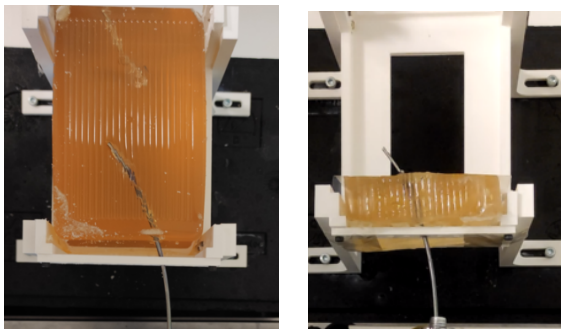


(b)

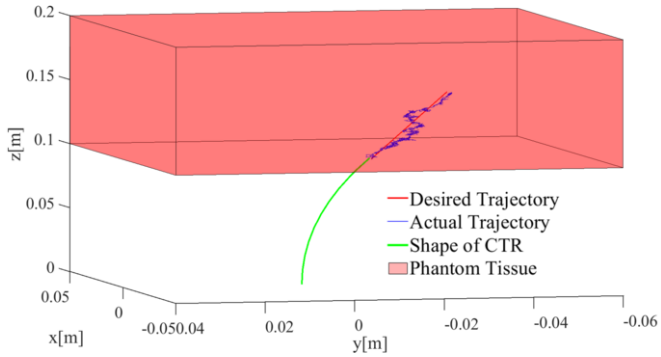
Figure 6.11: Experimental results for the 3rd scenario (S_3). The CTR is tasked to reach several target points shown by red circles while it is in contact with an unknown obstacle.

In the fourth scenario (S_4), the robot was navigated in the presence of a phantom tissue. We simulated two clinical scenarios. The first case study simulated percutaneous needle-based interventions such as prostate brachytherapy, where a needle is steered on a straight line to reach a deeply nested target in soft tissue [104]. The targets were selected inside the tissue within the robot's workspace, 65 mm from the entry point (Fig. 6.12(b)).

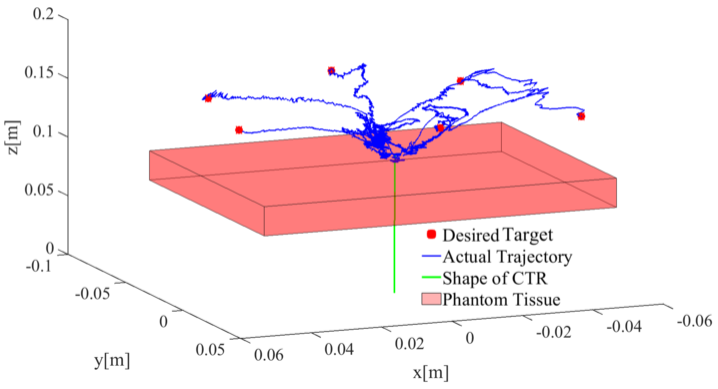
The 2^{nd} case study simulates percutaneous lung biopsy, where a needle is inserted through the skin, puncturing the pleura to reach the peripheral lung for sampling. Commonly, multiple incisions are used to reach multiple suspicious areas. Here, the robot is tasked to cut through a phantom tissue with a 25 mm thickness, simulating skin and pleura. Later it is tasked to reach 6 different points spread across its workspace from the same point of entry. We demonstrate that we can reach multiple sites via a single entry point (Fig. 6.12(b)). The phantom tissue in all the case studies is made following the recipe given in [105]. The tissue is made by mixing bovine gelatin powder with water at a temperature of 70°C. The weight ratio of gelatin-to-water in the mixture is 18% and the tissue's Young's modulus of elasticity was estimated to be 59 kPa. The elasticity of the synthetic tissues is similar to what is found in animal tissue. Although navigating the CTR inside a soft tissue is particularly challenging, the first case study shows that the robot is able to move along its tip direction following a straight line with a length of 65 mm inside the tissue (Fig. 6.12(a)). Figure 6.12(c) shows a representative result from this experiment. Over five insertions, the mean error of following the straight trajectory was 1.75 mm. As demonstrated by the second experiment, the robot's tip can reach target locations in different directions within its workspace despite making continuous contact with soft tissue. The robot was able to move its tip within 2 mm of the target in all 6 cases. (Fig. 6.12(d)).



(a) (b)



(c)



(d)

Figure 6.12: Experimental setup and results for the fourth scenario (S_4) (a) Robot's tip following a straight line while cutting through phantom tissue. (b) The robot cuts through the tissue to reach multiple target positions. (c) Results for robot cutting tissue along a straight line. (b) Results for robot's tip reaching 6 different target positions after cutting through tissue.

6.5 Concluding Remarks

In this chapter, we introduced a data-driven control strategy for the autonomous steering of CTRs. The control strategy relies only on a limited data set and is capable of rapidly adapting to the robot's nonlinear dynamics resulting in fast and accurate convergence. We studied the performance of the controller in extensive simulations and experiments. We considered a variety of scenarios, including the robot under external forces and the robot in contact with an obstacle. The proposed controller was capable of following a variety of trajectories within the robot's workspace with a maximum mean error of 2.4 mm at a frequency of 40 Hz and a velocity of 1 mm/s. Additionally, the controller was capable of steering the robot in the presence of unknown obstacles with a maximum error of 2 mm. Future work focuses on experimentally validating the proposed approach on CTRs with 3 tubes and investigating the application of the EDMD for controlling the robot's orientation in surgical tasks requiring higher dexterity.

Chapter 7

Conclusion

7.1 Summary

The main goal of this thesis was to investigate the use of CTRs as novel alternatives to needle-based interventions such as percutaneous lung biopsy. CTR is a unique type of robot that is well-suited for medical applications due to its small footprint and high level of dexterity. In recent years, CTRs have been proposed for a variety of minimally invasive surgical procedures. However, precise motion control in the presence of unknown interactions limits their application. Motivated by this, we explored data-driven control strategies for CTRs to achieve precise motion control within the context of distal lung sampling.

The conventional way to control CTRs is by solving their inverse kinematics based on their kinematic model. This approach has a couple of drawbacks. First, it is computationally expensive because it requires solving a set of differential equations with boundary conditions in order to obtain the robot's shape. Second, model-based solvers have difficulty adapting to unknown forces and uncertainties in the system. To overcome these difficulties, data-driven controllers are explored. Accordingly, two data-driven approaches are implemented: one that partly relies on the kinematic model, and one that is purely data-driven.

To test different control strategies without the need for physical experiments, a simulation environment is essential, where the computational cost of the kinematic model is crucial. Consequently, the first part of the thesis focuses on implementing the simulation environment and improving the kinematic model's computational efficiency. Two different approaches were taken to improve the kinematic model's performance. First, a nonlinear observer was proposed that can rapidly estimate a rod's shape without solving a boundary value problem. Second, approximations and optimizations were applied to the kinematic model in order to reduce its computational cost without compromising the accuracy of the model.

The second part of the thesis concentrates on data-driven control methods to provide accurate position control for CTRs in the presence of unknown interactions. Accordingly, two data-driven solutions are implemented. In the first approach, the model-based Jacobian is used as an initial input, and then it is updated in a similar manner to Broyden's method. This combines the advantages of model-based and data-driven control strategies. In the second

approach, Extended Dynamic Mode Decomposition (EDMD) is applied to approximate the Koopman operator. This holistic approach captures the nonlinear dynamics of the robot and its interactions with the environment. This purely data-driven approach allows accurate real-time control of the robot without the need for any prior knowledge of its model.

Finally, potential applications of CTRs are discussed in the context of needle-based interventions. Our results demonstrate that CTRs can be used as steerable needles and are suitable for Needle-Based Optical Endomicroscopy. To achieve these goals, a three-phase motion planner is proposed to determine the optimal entry point for a given task and to generate a stable path from the starting configuration to the entry point and then from the entry point to the target position.

7.2 Contributions

7.2.1 1^{st} contribution: computational improvements to the kinematic model of CTRs

According to the conventional approach, CTRs are modelled using the Cosserat rod theory. The Cosserat model contains a set of differential equations with boundary conditions split between the base and the tip of the rod. Boundary Value Problems (BVPs) must therefore be numerically solved, which can be a computationally expensive task. It has been demonstrated that it is possible to implement fast-enough solutions to the Cosserat model to control continuum robots. In spite of this, the computational cost of the model affects the performance and stability of model-based controllers. Consequently, less accurate models with low computational costs remain more appealing [69], [70]. Accordingly, the first contribution of the work is the creation of a computationally efficient simulation framework for CTRs. To achieve this, two different approaches are taken. The goal in both cases is to improve the computational efficiency of the kinematic model. First, a nonlinear observer is introduced to estimate the shape of the robot. Second, the existing model is implemented in C++ using approximations and optimizations. Both of the proposed solutions can be used to implement a real-time control system with conventional model-based controllers for CTRs with 2 tubes.

7.2.2 2^{nd} contribution: achieving precise position control in unknown environments using data-driven approaches

The second contribution of this work is achieving precise position control of CTRs in the presence of unknown obstacles and uncertainties in the system. The Cosserat model has an accuracy of approximately 9% of the robot's length. This error can be attributed to uncertainties in model parameters as well as neglecting torsion, shear strain, and friction. Additionally,

during minimally invasive surgery, the robot's body is constantly in contact with the surrounding environment. These unknown interactions cannot be taken into account by the kinematic model. Consequently, CTR control requires a controller that can adapt to unmodelled interaction forces and uncertainties in the system. To address the above challenges, we propose two data-driven solutions: a hybrid controller and a controller utilizing EDMD. Both of the controllers are capable of adapting to unknown external forces and providing real-time control for CTRs with 2 and 3 tubes.

7.2.3 3^{rd} contribution: demonstration of potential applications of CTRs in distal lung sampling via a three-phase motion planner

Finally, the thesis proposes potential applications of CTRs for distal lung sampling. Two potential applications are identified: using the robot as a steerable needle and performing optical endomicroscopy. Each application requires the tip of the robot to perform a movement in the direction in which its tip is pointing (local z-direction). In order to achieve this, a three-phase motion planner is introduced. During the first phase of the motion plan, all possible entry points for a given task are generated. Then, the second phase selects the most suitable entry point from the first phase, and a stable path between the selected configuration and the target is generated. Finally, the third phase generates a trajectory from the initial configuration to the entry point. These are done by the modified RRT* algorithm. As a final step experiments with phantom tissue are conducted to confirm the proposed applications.

7.3 Discussion

7.3.1 Data-driven Control Methods

There is a significant challenge in controlling CTRs, which is one of their main limitations. Due to the nonlinear dynamics of CTRs and unknown external forces in the system, model-based solutions are computationally expensive and have difficulty achieving precise position control.

Contrary to model-based solutions, Machine Learning (ML) methods offer the potential to overcome these limitations. However, most machine learning methods require large datasets. One example is the training of a deep neural network on a dataset of 100000 samples to learn the inverse kinematics of CTRs [46]. Unfortunately, it is often not feasible to collect large datasets of real robots, while the use of a simulation dataset for training does not guarantee satisfactory performance on real robots [47] mainly because it is difficult to simulate the robot's behaviour in contact with the environment or external forces in simulation. Consequently, previous works on this topic [46], [106], [45] have not been applied to the control of real CTRs and are tested only in simulations.

Since neither the model-based nor the learning-based approaches have provided satisfactory results, we decided to investigate possible data-driven approaches to achieve precise position control for CTRs. The difficulties mentioned above can be overcome by data-driven controllers that have the ability to adapt to the complex dynamics of the robot and its interactions with the environment in real time. Consequently, Chapter 4 proposes a hybrid control approach that combines the advantages of model-based and data-driven approaches. Then, a purely data-driven solution is proposed in Chapter 6. In the following section, we compare the experimental results of these control approaches.

7.3.2 Comparison of Data-driven Approaches

The hybrid controller receives the model-based prediction of the robot's differential kinematics and updates the Jacobian iteratively based on the feedback of the tip position received from the electromagnetic sensor. As the controller relies on the robot model, it first requires a system identification experiment to identify model parameters accurately, namely Young's and shear moduli of the tubes.

First, the EDMD accepts the system states and corresponding control inputs for a fixed period of time called DMD horizon. Next, it develops a linear map between control inputs and robot states in a higher dimensional space called lifted space. Finally, the robot's trajectory is controlled in this lifted space. We note that this is a continuous learning process. For instance, for a DMD horizon of 250 samples, at any given time instant the algorithm employs the previous 250 samples to predict the robot motion in the next sample time. Therefore, a change in external forces will affect the tip position, and the EDMD has the ability to adapt quickly and update the model accordingly. The proposed work is not linearizing the system in the sense that it neglects higher-order nonlinear terms. Instead, it is transforming the robot's dynamics into a higher dimensional space, where the mapping between inputs and outputs is linear. This will reduce the system's nonlinearity at the cost of increasing the system's dimensionality. Later, it is shown that the lifted space can be spanned by a set of orthonormal basis functions. The proposed EDMD approach estimates the basis functions to represent the lifted space as a linear state-space model suitable for model predictive control.

In order to compare the hybrid method with the EDMD, the following factors were taken into consideration: accuracy, computation time, learning curve, reliance on the kinematic model, and extendability to three tubes. Using either method, one can achieve accurate position control with a short learning curve, and it is possible to employ both methods on CTRs with three tubes. Fig. 7.1 shows a comparison between the tracking error of the hybrid controller and the controller using EDMD. As it can be seen, both controllers offer similar performance. Of note, in contrast to the hybrid controller, the proposed data-driven controller does not require any information from the model. Thus, obviating the need for modelling the robot and system identification with stereo cameras. Therefore, the proposed EDMD controller can

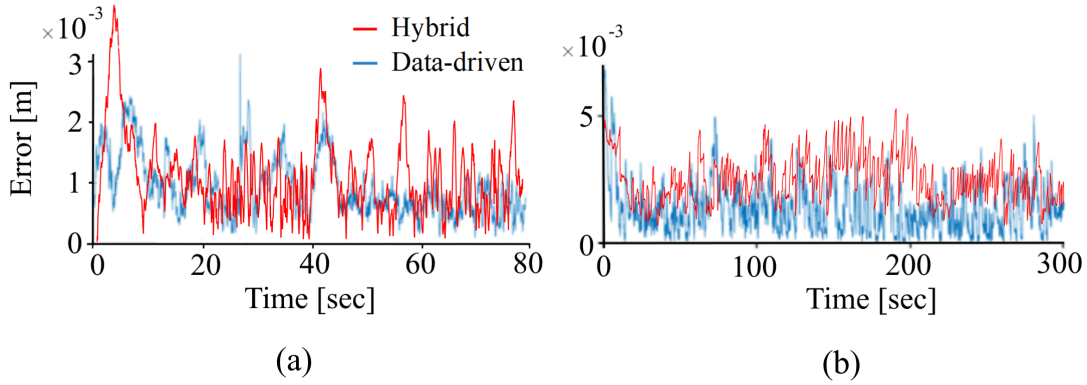


Figure 7.1: A comparison between the error of the proposed data-driven controller (blue line) and a hybrid controller (red line). (a) Following a square trajectory. (b) Following a long trajectory to the limits of robot workspace.

be deployed on any arbitrary CTR. In both cases, it takes less than 10 seconds to learn the robot's behaviour locally. A major advantage of the EDMD method is that it does not require an initial guess, whereas the hybrid method may not converge if the initial Jacobian is incorrect. The computational cost of the hybrid method is lower than that of the EDMD due to working with smaller matrices. The hybrid approach requires the pseudo-inverse of a 3x4 or 3x6 matrix depending on the number of tubes, while EDMD requires the singular value decomposition of a 16x250 or 18x250 matrix.

As a result, the EDMD represents a more advanced method that is capable of achieving similar results to the hybrid approach without the need to have any knowledge of the underlying system. Thus, it is a particularly attractive method for various continuum robot applications. The hybrid approach, on the other hand, is capable of providing similar accuracy, but it relies on the kinematic model and requires more tuning and parameterization compared to EDMD.

It is important to emphasize that the electromagnetic sensor has a position accuracy of 0.7 RMS (mm) with a maximum error of 1.8 mm. Thus, the tip position error is primarily caused by the inaccuracy of the sensor. In addition to this, the axes of the figures in this work often have different scales. According to this, both methods can achieve a very good accuracy, which can hardly be improved without increased computational cost or without changing the robot's design.

7.3.3 Distal Lung Sampling with CTRs

By utilizing the proposed data-driven control approaches, the challenges of using CTRs in constrained environments can be overcome, even if the robot is constantly in contact with its environment. As a result of these advancements, CTRs can become more appealing in minimally invasive surgeries. Thus, we examined the potential applications of CTRs in distal lung sampling. Accordingly, two potential applications for CTRs have been identified, namely optical endomicroscopy and steerable needles. Unfortunately, CTRs have limited workspace and are unstable in many configurations. Furthermore, it is also critical to move the robot's tip in its tip direction (local z-direction). Consequently, the application of CTRs in these scenarios requires a motion planner on top of the position controller. Therefore, we proposed a three-phase motion planner where the first phase generates all possible entry points (configurations) where movement in the local z-direction is possible. Then, from the generated entry points, the second phase selects the most suitable configuration for a specific task and provides a stable trajectory for the procedure. Following this, phase three generates a trajectory from the initial robot configuration to the selected entry point. Finally, the generated motion plans are experimentally verified using phantom tissues.

7.4 Future Directions

In this study, we demonstrated that CTRs can be used for optical endomicroscopy and as steerable needles. We achieved this by proposing data-driven control strategies and a three-phase motion planner. The proposed data-driven methods have the potential to overcome existing challenges associated with accurate position control of CTRs. Furthermore, the three-phase motion planner can provide motion plans for specific procedures. Although the results of this study are promising, there are still several challenges that can be addressed in order to make CTRs even more effective.

7.4.1 Pose Control of CTRs

A limitation of the proposed control strategies is that they do not take into account the orientation of the robot's tip. CTRs are commonly used as steerable needles to reach specific target points inside the body for therapeutic or diagnostic purposes. Although obtaining the desired target point is sufficient for these applications, achieving pose control would be of significant benefit and would extend the potential use of CTRs in minimally invasive surgery.

The CTR described in this work has 4 control inputs, two rotational and two translational inputs. With only four inputs, it is not always possible to control the robot's position and orientation (6 DoF). Even for CTRs with 3 tubes (6 inputs), accurate control of robot position and orientation at the same time is not always achievable, as the robot workspace contains

several singularities. One approach to overcome this problem is to use a specific tool with an articulated wrist that offers accurate orientation control [107]. As a result, the process of controlling the robot's Cartesian coordinates is decoupled from the process of controlling its orientation. Another approach would require the combination of complex robot design and advanced motion planning. In this case the robot design would optimize the robot parameters for specific applications while the motion planner would generate a feasible trajectory for the robot.

7.4.2 Experiments on a CTR with 3 Tubes

Throughout this work, simulations were carried out for CTRs with 3 tubes, but experiments were conducted only on CTRs with 2 tubes. Part of the reason for this is that the applications for CTRs suggested in Chapter 5 do not necessarily require a CTR with three tubes. Moreover, time constraints prevented us from carrying out experiments on CTRs with 3 tubes. Therefore, our future research will include testing data-driven control strategies and motion planners on CTRs with 3 tubes. In general, it is more challenging to control CTRs with 3 tubes; the number of unstable configurations and computation time increases with the number of tubes. In spite of this, the simulation results are promising; both data-driven strategies have been successfully extended to 3 tubes.

7.4.3 Including the Shape of the Robot to Position and Motion Control

To achieve data-driven control and motion planning, only the tip position of the robot was considered in this work. The performance of the data-driven methods may be further improved by including more points along the shape of the robot in the feedback loop. On top of this, the motion planner could generate more realistic trajectories by taking the shape of the robot into account.

Bibliography

[1] N. Krstajić, B. Mills, I. Murray, A. Marshall, D. Norberg, T. H. Craven, P. Emanuel, T. R. Choudhary, G. O. Williams, E. Scholefield *et al.*, "Low-cost high sensitivity pulsed endomicroscopy to visualize tricolor optical signatures," *Journal of biomedical optics*, vol. 23, no. 7, p. 076005, 2018.

[2] S. Blandin Knight, P. A. Crosbie, H. Balata, J. Chudziak, T. Hussell, and C. Dive, "Progress and prospects of early detection in lung cancer," *Open biology*, vol. 7, no. 9, p. 170070, 2017.

[3] J. A. Engh, D. S. Minhas, D. Kondziolka, and C. N. Riviere, "Percutaneous intracerebral navigation by duty-cycled spinning of flexible bevel-tipped needles," *Neurosurgery*, vol. 67, no. 4, pp. 1117–1123, 2010.

[4] J. Burgner, P. J. Swaney, R. A. Lathrop, K. D. Weaver, and R. J. Webster, "Debulking from within: a robotic steerable cannula for intracerebral hemorrhage evacuation," *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 9, pp. 2567–2575, 2013.

[5] Z. Mitros, S. Sadati, C. Seneci, E. Bloch, K. Leibrandt, M. Khadem, L. Da Cruz, and C. Bergeles, "Optic nerve sheath fenestration with a multi-arm continuum robot," *IEEE robotics and automation letters*, vol. 5, no. 3, pp. 4874–4881, 2020.

[6] A. H. Gosline, N. V. Vasilyev, E. J. Butler, C. Folk, A. Cohen, R. Chen, N. Lang, P. J. Del Nido, and P. E. Dupont, "Percutaneous intracardiac beating-heart surgery using metal mems tissue approximation tools," *The International journal of robotics research*, vol. 31, no. 9, pp. 1081–1093, 2012.

[7] D. Thomas, E. P. Scholz, P. A. Schweizer, H. A. Katus, and R. Becker, "Initial experience with robotic navigation for catheter ablation of paroxysmal and persistent atrial fibrillation," *Journal of electrocardiology*, vol. 45, no. 2, pp. 95–101, 2012.

[8] P. J. Swaney, A. W. Mahoney, A. A. Remirez, E. Lamers, B. I. Hartley, R. H. Feins, R. Alterovitz, and R. J. Webster, "Tendons, concentric tubes, and a bevel tip: Three steerable robots in one transoral lung access system," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 5378–5383.

[9] Z. Mitros, S. H. Sadati, R. Henry, L. Da Cruz, and C. Bergeles, "From theoretical work to clinical translation: Progress in concentric tube robots," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, pp. 335–359, 2022.

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

[11] P. E. Dupont, J. Lock, B. Itkowitz, and E. Butler, "Design and control of concentric-tube robots," *IEEE Transactions on Robotics*, vol. 26, no. 2, pp. 209–225, 2009.

[12] D. C. Rucker, B. A. Jones, and R. J. Webster III, "A geometrically exact model for externally loaded concentric-tube continuum robots," *IEEE Transactions on Robotics*, vol. 26, no. 5, pp. 769–780, 2010.

[13] S. Hirose, "Biologically inspired robots," *Snake-Like Locomotors and Manipulators*, 1993.

[14] G. S. Chirikjian and J. W. Burdick, "A modal approach to hyper-redundant manipulator kinematics," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 3, pp. 343–354, 1994.

[15] G. Chirikjian and J. W. Burdick, "Kinematically optimal hyper-redundant manipulator configurations," *IEEE transactions on Robotics and Automation*, vol. 11, no. 6, pp. 794–806, 1995.

[16] I. A. Gravagne, C. D. Rahn, and I. D. Walker, "Large deflection dynamics and control for planar continuum robots," *IEEE/ASME transactions on mechatronics*, vol. 8, no. 2, pp. 299–307, 2003.

[17] I. A. Gravagne and I. D. Walker, "Manipulability, force, and compliance analysis for planar continuum manipulators," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 3, pp. 263–273, 2002.

[18] M. W. Hannan and I. D. Walker, "Kinematics and the implementation of an elephant's trunk manipulator and other continuum style robots," *Journal of robotic systems*, vol. 20, no. 2, pp. 45–63, 2003.

[19] N. 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*. IEEE, 2005, pp. 3012–3017.

[20] J. Bonatti, G. Vetrovec, C. Riga, O. Wazni, and P. Stadler, "Robotic technology in cardiovascular medicine," *Nature Reviews Cardiology*, vol. 11, no. 5, p. 266, 2014.

[21] H. Rafii-Tari, C. J. Payne, and G.-Z. Yang, "Current and emerging robot-assisted endovascular catheterization technologies: a review," *Annals of biomedical engineering*, vol. 42, no. 4, pp. 697–715, 2014.

[22] E. M. Boctor, M. A. Choti, E. C. Burdette, and R. J. Webster Iii, "Three-dimensional ultrasound-guided robotic needle placement: an experimental evaluation," *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 4, no. 2, pp. 180–191, 2008.

[23] R. J. Hendrick, C. R. Mitchell, S. D. Herrell, and R. J. Webster III, "Hand-held transendoscopic robotic manipulators: A transurethral laser prostate surgery case study," *The International journal of robotics research*, vol. 34, no. 13, pp. 1559–1572, 2015.

[24] R. E. Goldman, A. Bajo, L. S. MacLachlan, R. Pickens, S. D. Herrell, and N. Simaan, "Design and performance evaluation of a minimally invasive telerobotic platform for transurethral surveillance and intervention," *IEEE transactions on biomedical engineering*, vol. 60, no. 4, pp. 918–925, 2012.

[25] J. Burgner, D. C. Rucker, H. B. Gilbert, P. J. Swaney, P. T. Russell, K. D. Weaver, and R. J. Webster, "A telerobotic system for transnasal surgery," *IEEE/ASME Transactions on Mechatronics*, vol. 19, no. 3, pp. 996–1006, 2013.

[26] H.-S. Yoon, H.-J. Cha, J. Chung, and B.-J. Yi, "Compact design of a dual master-slave system for maxillary sinus surgery," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 5027–5032.

[27] M. Khadem, J. O'Neill, Z. Mitros, L. d. Cruz, and C. Bergeles, "Autonomous steering of concentric tube robots for enhanced force/velocity manipulability," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 2197–2204.

[28] D. C. Rucker and R. J. Webster, "Computing jacobians and compliance matrices for externally loaded continuum robots," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 945–950.

[29] K. Leibrandt, C. Bergeles, and G.-Z. Yang, "Concentric tube robots: Rapid, stable path-planning and guidance for surgical use," *IEEE Robotics & Automation Magazine*, vol. 24, no. 2, pp. 42–53, 2017.

[30] H. B. Gilbert, J. Neimat, and R. J. Webster, "Concentric tube robots as steerable needles: Achieving follow-the-leader deployment," *IEEE Transactions on Robotics*, vol. 31, no. 2, pp. 246–258, 2015.

[31] A. V. Kudryavtsev, M. T. Chikhaoui, A. Liadov, P. Rougeot, F. Spindler, K. Rabenorosoa, J. Burgner-Kahrs, B. Tamadazte, and N. Andreff, "Eye-in-hand visual servoing of concentric tube robots," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2315–2321, 2018.

[32] M. Khadem, J. O'Neill, Z. Mitros, L. da Cruz, and C. Bergeles, "Autonomous steering of concentric tube robots via nonlinear model predictive control," *IEEE Transactions on Robotics*, vol. 36, no. 5, pp. 1595–1602, 2020.

[33] J. Till, V. Aloi, K. E. Riojas, P. L. Anderson, R. J. Webster III, and C. Rucker, "A dynamic model for concentric tube robots," *IEEE Transactions on Robotics*, vol. 36, no. 6, pp. 1704–1718, 2020.

[34] G. Fagogenis, M. Mencattelli, Z. Machaidze, B. Rosa, K. Price, F. Wu, V. Weixler, M. Saeed, J. E. Mayer, and P. E. Dupont, "Autonomous robotic intracardiac catheter navigation using haptic vision," *Science Robotics*, vol. 4, no. 29, p. eaaw1977, 2019.

[35] A. Razjigaev, R. Crawford, J. Roberts, and L. Wu, "Teleoperation of a concentric tube robot through hand gesture visual tracking," in *IEEE Int. Conf. on Robotics and Biomimetics*, 2017, pp. 1175–1180.

[36] J. Burgner, D. C. Rucker, H. B. Gilbert, P. J. Swaney, P. T. Russell, K. D. Weaver, and R. J. Webster, "A telerobotic system for transnasal surgery," *IEEE/ASME Transactions on Mechatronics*, vol. 19, no. 3, pp. 996–1006, 2014.

[37] H. Donat, S. Lilge, J. Burgner-Kahrs, and J. J. Steil, "Estimating tip contact forces for concentric tube continuum robots based on backbone deflection," *IEEE Transactions on Medical Robotics and Bionics*, vol. 2, no. 4, pp. 619–630, 2020.

[38] L. G. Torres, C. Baykal, and R. Alterovitz, "Interactive-rate motion planning for concentric tube robots," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 1915–1921.

[39] A. Kuntz, M. Fu, and R. Alterovitz, "Planning high-quality motions for concentric tube robots in point clouds via parallel sampling and optimization," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 2205–2212.

[40] K. Leibrandt, C. Bergeles, and G.-Z. Yang, "Concentric tube robots: Rapid, stable path-planning and guidance for surgical use," *IEEE Robotics Automation Magazine*, vol. 24, no. 2, pp. 42–53, 2017.

[41] J. M. Bern, Y. Schnider, P. Banzet, N. Kumar, and S. Coros, "Soft robot control with a learned differentiable model," in *2020 3rd IEEE International Conference on Soft Robotics (RoboSoft)*. IEEE, 2020, pp. 417–423.

[42] D. Bruder, C. D. Remy, and R. Vasudevan, "Nonlinear system identification of soft robot dynamics using koopman operator theory," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 6244–6250.

[43] R. Morimoto, S. Nishikawa, R. Niiyama, and Y. Kuniyoshi, "Model-free reinforcement learning with ensemble for a soft continuum robot arm," in *2021 IEEE 4th International Conference on Soft Robotics (RoboSoft)*. IEEE, 2021, pp. 141–148.

[44] D. Büchler, S. Guist, R. Calandra, V. Berenz, B. Schölkopf, and J. Peters, "Learning to play table tennis from scratch using muscular robots," *IEEE Transactions on Robotics*, 2022.

[45] K. Iyengar, G. Dwyer, and D. Stoyanov, "Investigating exploration for deep reinforcement learning of concentric tube robot control." *International Journal of Computer Assisted Radiology and Surgery*, 2020.

[46] N. Liang, R. M. Grassmann, S. Lilge, and J. Burgner-Kahrs, "Learning-based inverse kinematics from shape as input for concentric tube continuum robots," in *IEEE Int. Conf. on Robotics and Automation*, 2021, pp. 1387–1393.

[47] G. Fang, Y. Tian, Z.-X. Yang, J. M. Geraedts, and C. C. Wang, "Efficient jacobian-based inverse kinematics with sim-to-real transfer of soft robots by learning," *IEEE/ASME Transactions on Mechatronics*, 2022.

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

[49] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[50] S. M. LaValle and J. J. Kuffner Jr, "Randomized kinodynamic planning," *The international journal of robotics research*, vol. 20, no. 5, pp. 378–400, 2001.

[51] S. Karaman, M. R. Walter, A. Perez, E. Frazzoli, and S. Teller, "Anytime motion planning using the rrt," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 1478–1483.

[52] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The international journal of robotics research*, vol. 30, no. 7, pp. 846–894, 2011.

[53] C. Bergeles and P. E. Dupont, "Planning stable paths for concentric tube robots," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 3077–3082.

[54] L. G. Torres, C. Baykal, and R. Alterovitz, "Interactive-rate motion planning for concentric tube robots," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 1915–1921.

[55] K. Wu, L. Wu, and H. Ren, "Motion planning of continuum tubular robots based on centerlines extracted from statistical atlas," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 5512–5517.

[56] A. Kuntz, M. Fu, and R. Alterovitz, "Planning high-quality motions for concentric tube robots in point clouds via parallel sampling and optimization," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 2205–2212.

[57] W. Sun, L. G. Torres, J. v. d. Berg, and R. Alterovitz, "Safe motion planning for imprecise robotic manipulators by minimizing probability of collision," in *Robotics Research*. Springer, 2016, pp. 685–701.

[58] D. C. Rucker and R. J. Webster, "Statics and dynamics of continuum robots with general tendon routing and external loading," *IEEE Trans. Robotics*, vol. 27, no. 6, pp. 1033–1044, 2011.

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

[60] D. C. Rucker, B. A. Jones, and R. J. Webster, "A geometrically exact model for externally loaded concentric-tube continuum robots," *IEEE Trans. Robotics*, vol. 26, no. 5, pp. 769–780, 2010.

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

[62] F. Renda, M. Cianchetti, H. Abidi, J. Dias, and L. Seneviratne, "Screw-Based Modeling of Soft Manipulators With Tendon and Fluidic Actuation," *Journal of Mechanisms and Robotics*, vol. 9, no. 4, 05 2017.

[63] B. Thamo, K. Dhaliwal, and M. Khadem, "Rapid solution of cosserat rod equations via a nonlinear partial observer," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 9433–9438.

[64] E. Mackute, B. Thamo, K. Dhaliwal, and M. Khadem, "Shape estimation of concentric tube robots using single point position measurement," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 3972–3978.

[65] J. Till, C. E. Bryson, S. Chung, A. Orekhov, and D. C. Rucker, "Efficient computation of multiple coupled cosserat rod models for real-time simulation and control of parallel continuum manipulators," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 5067–5074.

[66] R. Xu, A. Asadian, S. F. Atashzar, and R. V. Patel, "Real-time trajectory tracking for externally loaded concentric-tube robots," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 4374–4379.

[67] M. Khadem, J. O'Neill, Z. Mitros, L. da Cruz, and C. Bergeles, "Autonomous steering of concentric tube robots via nonlinear model predictive control," *IEEE Transactions on Robotics*, pp. 1–8, 2020.

[68] M. Khadem, J. O'Neill, Z. Mitros, L. d. Cruz, and C. Bergeles, "Autonomous steering of concentric tube robots for enhanced force/velocity manipulability," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 2197–2204.

[69] A. Bajo and N. Simaan, "Hybrid motion/force control of multi-backbone continuum robots," *The International Journal of Robotics Research*, vol. 35, no. 4, pp. 422–434, 2016.

[70] M. T. Chikhaoui, S. Lilge, S. Kleinschmidt, and J. Burgner-Kahrs, "Comparison of modeling approaches for a tendon actuated continuum robot with three extensible segments," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 989–996, 2019.

[71] Z. Mitros, M. Khadem, C. Seneci, S. Ourselin, L. Da Cruz, and C. Bergeles, "Towards modelling multi-arm robots: Eccentric arrangement of concentric tubes," in *2018 7th IEEE International Conference on Biomedical Robotics and Biomechatronics (Biorob)*, 2018, pp. 43–48.

[72] H. B. Gilbert and *et al.*, "Elastic stability of concentric tube robots: A stability measure and design test," *IEEE Trans. Robotics*, vol. 32, no. 1, pp. 20–35, 2016.

[73] R. Xu, S. F. Atashzar, and R. V. Patel, "Kinematic instability in concentric-tube robots: Modeling and analysis," in *5th IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics*, 2014, pp. 163–168.

[74] M. A. M. Cheema, J. E. Fletcher, D. Xiao, and M. F. Rahman, "A linear quadratic regulator-based optimal direct thrust force control of linear permanent-magnet synchronous motor," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 5, pp. 2722–2733, 2016.

[75] F. Callier and J. Willems, "Criterion for the convergence of the solution of the riccati differential equation," *IEEE Transactions on Automatic Control*, vol. 26, no. 6, pp. 1232–1242, 1981.

[76] J. Nocedal and S. Wright, *Interior-Point Methods for Nonlinear Programming*. New York, NY: Springer New York, 2006, pp. 563–597.

[77] F. E. Curtis and X. Que, "A quasi-Newton algorithm for nonconvex, nonsmooth optimization with global convergence guarantees," *Mathematical Programming Computation*, vol. 7, no. 4, pp. 399–428, 2015.

[78] M. J. D. Powell, "On search directions for minimization algorithms," *Mathematical Programming*, vol. 4, no. 1, pp. 193–201, 1973.

[79] P. Bogacki and L. Shampine, "A 3(2) pair of Runge - Kutta formulas," *Applied Mathematics Letters*, vol. 2, no. 4, pp. 321–325, 1989.

[80] K. Ahnert and M. Mulansky, "Odeint–solving ordinary differential equations in c++," in *AIP Conference Proceedings*, vol. 1389, no. 1. American Institute of Physics, 2011, pp. 1586–1589.

[81] B. Thamo, F. Alambeigi, K. Dhaliwal, and M. Khadem, "A hybrid dual jacobian approach for autonomous control of concentric tube robots in unknown constrained environments," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 2809–2815.

[82] C. G. Broyden, "A class of methods for solving nonlinear simultaneous equations," *Mathematics of computation*, vol. 19, no. 92, pp. 577–593, 1965.

[83] F. Alambeigi, Z. Wang, R. Hegeman, Y.-H. Liu, and M. Armand, "Autonomous data-driven manipulation of unknown anisotropic deformable tissues using unmodelled continuum manipulators," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 254–261, 2019.

[84] F. Alambeigi, Z. Wang, Y.-H. Liu, R. H. Taylor, and M. Armand, "A versatile data-driven framework for model-independent control of continuum manipulators interacting with obstructed environments with unknown geometry and stiffness," *arXiv preprint arXiv:2005.01951*, 2020.

[85] F. Alambeigi, Z. Wang, R. Hegeman, Y.-H. Liu, and M. Armand, "A robust data-driven approach for online learning and manipulation of unmodeled 3-d heterogeneous compliant objects," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4140–4147, 2018.

[86] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: modelling, planning and control*. Springer Science & Business Media, 2010.

[87] A. M. Okamura, C. Simone, and M. D. O'leary, "Force modeling for needle insertion into soft tissue," *IEEE transactions on biomedical engineering*, vol. 51, no. 10, pp. 1707–1716, 2004.

[88] J. Zhang, K. A. Gold, H. Y. Lin, S. G. Swisher, Y. Xing, J. J. Lee, E. S. Kim, and W. N. William Jr, "Relationship between tumor size and survival in non-small-cell lung cancer (NSCLC): an analysis of the surveillance, epidemiology, and end results (SEER) registry," *Journal of thoracic oncology : official publication of the International Association for the Study of Lung Cancer*, vol. 10, no. 4, pp. 682–690, apr 2015.

[89] C. Yang, Y. Xie, S. Liu, and D. Sun, "Force Modeling, Identification, and Feedback Control of Robot-Assisted Needle Insertion: A Survey of the Literature." *Sensors (Basel, Switzerland)*, vol. 18, no. 2, feb 2018.

[90] S. Zuo and G.-Z. Yang, "Endomicroscopy for computer and robot assisted intervention," *IEEE Reviews in Biomedical Engineering*, vol. 10, pp. 12–25, 2017.

[91] M. Khadem, C. Rossa, R. S. Sloboda, N. Usmani, and M. Tavakoli, "Mechanics of tissue cutting during needle insertion in biological tissue," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 800–807, 2016.

[92] H. B. Gilbert, R. J. Hendrick, and R. J. Webster III, "Elastic stability of concentric tube robots: A stability measure and design test," *IEEE Transactions on Robotics*, vol. 32, no. 1, pp. 20–35, 2015.

[93] P. Ardón, È. Pairet, K. S. Lohan, S. Ramamoorthy, and R. Petrick, "Affordances in robotic tasks–a survey," *arXiv preprint arXiv:2004.07400*, 2020.

[94] M. Khadem, L. Da Cruz, and C. Bergeles, "Force/velocity manipulability analysis for 3d continuum robots," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 4920–4926.

[95] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-d point sets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, no. 5, pp. 698–700, 1987.

[96] B. Thamo, D. Hanley, K. Dhaliwal, and M. Khadem, "Data-driven steering of concentric tube robots in unknown environments via dynamic mode decomposition," *IEEE Robotics and Automation Letters*, 2022.

[97] B. O. Koopman, "Hamiltonian systems and transformation in hilbert space," *Proceedings of the National Academy of Sciences*, vol. 17, no. 5, pp. 315–318, 1931.

[98] S. E. Otto and C. W. Rowley, “Koopman operators for estimation and control of dynamical systems,” *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, no. 1, pp. 59–87, 2021.

[99] M. Korda and I. Mezić, “Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control,” *Automatica*, vol. 93, pp. 149–160, 2018.

[100] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley, “A data–driven approximation of the koopman operator: Extending dynamic mode decomposition,” *J. of Nonlinear Science*, vol. 25, no. 6, pp. 1307–1346, 2015.

[101] J. L. Proctor, S. L. Brunton, and J. N. Kutz, “Generalizing koopman theory to allow for inputs and control,” *SIAM Journal on Applied Dynamical Systems*, vol. 17, no. 1, pp. 909–930, 2018.

[102] P. J. L., B. S. L., and J. N. Kutz, “Dynamic mode decomposition with control,” *SIAM Journal on Applied Dynamical Systems*, vol. 15, no. 1, pp. 142–161, 2016.

[103] M. Kögel and R. Findeisen, “A fast gradient method for embedded linear predictive control,” *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 1362–1367, 2011, 18th IFAC World Congress.

[104] M. Khadem and *et al.*, “Robotics-assisted needle steering around anatomical obstacles using notched steerable needles,” *IEEE J. Biomedical and Health Informatics*, pp. 1–1, 2017.

[105] M. Khadem, C. Rossa, N. Usmani, R. S. Sloboda, and M. Tavakoli, “A two-body rigid/flexible model of needle steering dynamics in soft tissue,” *IEEE/ASME Trans. on Mechatronics*, vol. 21, no. 5, pp. 2352–2364, 2016.

[106] K. Iyengar and D. Stoyanov, “Deep reinforcement learning for concentric tube robot control with a goal-based curriculum,” in *IEEE Int. Conf. on Robotics and Automation*, 2021, pp. 1459–1465.

[107] P. A. York, P. J. Swaney, H. B. Gilbert, and R. J. Webster, 3rd, “A wrist for Needle-Sized surgical robots,” *IEEE Int Conf Robot Autom*, vol. 2015, pp. 1776–1781, May 2015.