UNIVERSITY OF CALIFORNIA SAN DIEGO

A 3D Haptic Trackball For Teleoperating Tendon-Driven Continuum Robots

A thesis submitted in partial satisfaction of the
requirements for the degree of Master of Science

in

Engineering Sciences (Mechanical Engineering)

by

Mufeng Xie

Committee in charge:

Professor Tania Morimoto, Chair
Professor Nicholas Gravish
Professor Michael Tolley

2021

The Thesis of Mufeng Xie is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2021

DEDICATION

For my parents.

TABLE OF CONTENTS

Thesis Approval Page iii

Dedication iv

Table of Contents v

List of Figures vii

List of Tables x

Acknowledgements xi

Abstract of the Thesis xii

Chapter 1 Introduction 1
- 1.1 Continuum robots 3
- 1.2 Tendon-driven robots 5
- 1.3 Conventional interfaces of surgical robots 6
- 1.4 Haptic interfaces 8
- 1.5 Contributions 10

Chapter 2 System Design 12
- 2.1 Hardware design 13
- 2.2 Haptic feedback 15

Chapter 3 Teleoperation Methods 19
- 3.1 Linear insertion 20
- 3.2 Inverse kinematics 22
- 3.3 Mappings in 3D space 28
- 3.4 Camera-based vision tracking 29
- 3.5 Software implementation 31

Chapter 4 User Study 34
- 4.1 Study design 34
  - 4.1.1 Device setups 34
  - 4.1.2 Training section 35
  - 4.1.3 Following paths task 35
  - 4.1.4 Avoiding obstacles task 37
  - 4.1.5 Other preparations for the study 39
- 4.2 Experiment setup 41
- 4.3 Evaluation metrics 44
- 4.4 Study results 45

Chapter 5 Conclusions and Future Work 53

Bibliography 56

LIST OF FIGURES

Figure 1.1. This figure, originally appeared in Jessica Burgner-Kahrs et al.[18], shows the number of DOFs from traditional rigid links robots to continuum robots(Images: top copyright from 2015 Kuka Robotics Corp.;bottom copyright from 2015 Hansen Medical Inc.) 4

Figure 1.2. This figure, originally appeared in Takahisa Kato et al.[19], shows a tendon-driven robot with a diameter of 3.4mm 6

Figure 1.3. This figure, originally appeared in Matthew Csencsits et al.[22], shows a joystick interface for controlling an Air-Octor continuum robot arm. Besides the mapping from the motion of joystick arm to robot tip position, there are several switches for changing control modes. 7

Figure 1.4. This figure, originally appeared in Hyun-Soo Yoon et al.[37], shows an intuitive interface for controlling a multi-moduled continuum robot. The interface, which is held by human hands, can not only control the tip position but also the body shape of the robot. 8

Figure 1.5. This figure, originally appeared in Yasuyoshi Yokokohji et al.[35], shows a typical haptic interface interacting with virtual environments. 9

Figure 1.6. This figure, originally appeared in Bo Ouyang et al.[4], shows a Phantom Omni used as a input device. 10

Figure 2.1. An overview of our 3D haptic trackball interface with dimensions. 13

Figure 2.2. Motors and shafts in the 2D part of the system. 14

Figure 2.3. The linear capstan-drive connected with the third motor. 15

Figure 2.4. This figure shows the haptic feedback of a virtual wall. 16

Figure 2.5. The mechanism of force feedback on the trackball for a single motor shaft. The other motor shaft is oriented perpendicularly to the motor shown in this figure. 17

Figure 2.6. The mechanism of force feedback on the linear capstan-drive. 18

Figure 3.1. The layout of tendons going through spacer disks in two sections: (a) the first section with 6 tendons, (b) the second section with 3 tendons. 20

Figure 3.2. The backbone and driving components of the tendon-driven robot. 21

Figure 3.3. The tendon-driven robot with linear actuator as a whole system. 22

Figure 3.4. This figure, originally appeared in Robert J. Webster III et al. [33], shows the mappings in continuum robots between actuator space, configuration space and task space. 23

Figure 3.5. This figure, originally appeared in Robert J. Webster III et al. [33], shows the single section model of a continuum robot. (a) When ϕ is zero, the robot is in the x-z plane. (b) As ϕ changes, the arc rotates along z-axis. . . 24

Figure 3.6. Compensation of coupled motion on z axis: (a) Without compensation, (b) With compensation 28

Figure 3.7. The 3D printed sphere painted in red for color tracking. 30

Figure 3.8. The red tip in camera view and its bounding box recognized by blob analysis. 31

Figure 3.9. The workflow of the signals in the system. 1: User input: $u = (x, y, z_{leader})$. 2: Encoder reading (x, y). 3: Tendon lengths q. 4: Actual tip position (x_a, y_a, z_a). 5: Actual tip position (x_a, y_a, z_a) and coupled z-displacement $z_{coupled}$. 6: Force feedback F 7: Final z-displacement $z = z_{follower} - z_{coupled}$. 32

Figure 4.1. The virtual environment in MATLAB. The user will use the haptic trackball interface to control the sphere and hit the virtual walls to feel the force feedback. 35

Figure 4.2. The layout of following path task. 36

Figure 4.3. The force feedback in following paths task. 37

Figure 4.4. The layout of avoiding obstacle task. 38

Figure 4.5. The force feedback in avoiding obstacle task. 39

Figure 4.6. A user teleoperating the robot with our proposed interface. 41

Figure 4.7. The workspace. Two cameras are mounted on the right wall and top roof. 42

Figure 4.8. The mapping of the system. On the top is the frame of motion of the tendon-driven robot. Bottom left and bottom right show the mapping of joystick and trackball interface. 43

Figure 4.9. The box plot of errors in following paths task. 46

Figure 4.10. The actual paths using trackball with force feedback. 47

Figure 4.11. The actual paths using trackball without force feedback. 47

Figure 4.12. The actual paths using joystick. 48

Figure 4.13. The box plot of runtime in following paths task. 48

Figure 4.14. The box plot of runtime in avoiding obstacles task. 50

LIST OF TABLES

Table 3.1. Variables used for inverse kinematics. 25

Table 4.1. Order of tasks and device setups. For device setups, A = trackball with force feedback, B = Trackball without force feedback, C = Joystick. 40

Table 4.2. Error (mm) and runtime (s) of 3 device setups in following paths task. . . . 49

Table 4.3. Average times hitting obstacles in each trial and runtime of avoiding obstacles task. 50

Table 4.4. Average ratings of 3 device setups in different aspects, with 5 being the best (or most mental effort) and 1 being the worst (or least mental effort). 51

Table 4.5. Average ranking of 3 device setups in different aspects, with 1 being the first rank and 3 being the lowest rank. 52

ACKNOWLEDGEMENTS

First, I would like to thank Professor Morimoto in particular for giving me the research opportunity in Morimoto Lab. She has always been supportive and encouraging since I joined the lab. She has not only provided me with guidance and inspirations on the projects, but also been a great mentor for me as I go through difficulties in both research and life. It has been a valuable experience for me to learn and work in Morimoto Lab.

In addition, I would like to thank the other committee members: Professor Michael Tolley and Professor Nick Gravish, whose outstanding research works in robotics inspired me a lot in my graduate study. It is my pleasure to have them as my thesis committee members.

Besides, I would like to thank Dr.Cedric Girerd for his persistent support on this project. Further, I would like to thank all the members in Morimoto Lab for providing useful advice and coordinating with me during my experiments. Finally, I would also like to thank all the participants of the user study for participating and providing experimental data.

ABSTRACT OF THE THESIS

A 3D Haptic Trackball For Teleoperating Tendon-Driven Continuum Robots

by

Mufeng Xie

Master of Science in Engineering Sciences (Mechanical Engineering)

University of California San Diego, 2021

Professor Tania Morimoto, Chair

Continuum robots are commonly used as surgical robots, due to their flexibility in confined workspaces. Since continuum robots have a very large number of degrees of freedom, it is a challenging task for humans to control them, and a proper user interface is required to serve as a input device. Haptic interfaces offer significant potential to be used as input devices for teleoperation, due to their force feedback that can help users better control the robots. However, few haptic interface systems have been developed for the control of continuum robots. We proposed a novel 3D haptic trackball as an input device to control tendon-driven continuum robots. The system will let the user control the tip position of a tendon-driven robot with high accuracy, as well as provide haptic force feedback when necessary, so as to assist the user with controlling the robot in different tasks. We design a user study to evaluate the performance of our proposed system and compare it with an off-the-shelf interface, a joystick on a gamepad, which has been previously used as an experimental interface for continuum robots. We asked the users to perform a path following task and a obstacle avoiding task in a 600mm × 300mm × 300mm workspace. Results show that our proposed haptic trackball interface enables higher accuracy in path following and lower chances of hitting obstacles in obstacle avoiding, compared to using the joystick interface. Although the runtime of our system in each task is longer than that of the joystick, considering the scenario of surgical processes, where accuracy and safety are often much more important than speed, the sacrifice in speed has a reduced influence on system performance. Overall, our proposed interface performs better than conventional interfaces in terms of positional precision and obstacles avoidance, thus having great potentials to be used in robotic assisted surgery.

Chapter 1

Introduction

Over the past decade, the field of robotics has seen great improvements in numerous applications. Robots are no longer simple servers in daily life or experimental tools used only in laboratories. Instead, robots nowadays are intelligent enough to work interactively with human in more complicated scenarios. Benefiting from the fast-growing techniques in robotics, robotic assisted surgery has also seen significant progress. For surgical robots, precision, ability to access surgical sites, and minimal harm to the human body are always desired [18]. Continuum robots help meet these requirements due to their curvilinear and flexible shape and infinite degrees of freedom [18], [8]. They can reach targeted sites, such as small incisions or orifices, with complex paths in constrained spaces within the human body and can complete delicate surgical operations with their miniaturized end effectors [8].

A representative example of continuum robot is a tendon-driven robot, which consists of an elastic backbone with spacer disks. Tendons are arranged along the circumferences of the spacer disks, with equal angles apart. By controlling the length of each tendon, the robot backbone can be deployed with different curvatures. The length of each section is fixed while the curvatures change, and a design of small section length can realize a large bending radii of the robot. Tendon-driven robots work particularly well in endoscopy, such as neuroendoscopy, as they can provide adequate power through curved paths in constrained spaces, while at the same time, keeping a safe distance between the actuators and patients [19].

While some simple instruments with tendon-driven mechanisms are usually operated manually, a robotized teleoperation interface is preferred when it comes to higher DOF motions in catheters or endoscopes [19] [7] [5]. Although there are plenty of studies on the kinematic model and control model of continuum robots, research on design of such interface systems is limited. A joystick or gamepad is used most often as an input device for continuum robots [22] [6]. Joysticks are easy to use and the mapping for teleoperation is straightforward. Some recent studies are trying to build a more intuitive interface, which mimics the kinematic structure of the continuum robots that they intend to control remotely [37]. A novel design even uses leap motion based on an optical hand tracking system as a user interface in robotic endonasal surgery [32]. However, neither method provides enough feedback to the user, which means the user can have a difficult time adjusting the input motion corresponding to the robot output in realtime.

To enable the realtime adjustment of user inputs, we aim for a haptic interface capable of providing force feedback to the user. Research in haptics shows that the operator's stress and task errors can be reduced by using haptic devices which allow the user to perceive force feedback from the working environment [11]. There are some studies using haptic interfaces as input devices for continuum robots, and these typically use conventional haptic devices, such as the Novint Falcon [6]. However, most haptic devices have a limited workspace due to the kinematic structure, so the mappings are usually from the position of the input device to the velocity of robot (P2V). P2V mappings work well with high-speed motions, but are less effective in achieving accurate robot positions. In contrast, the mapping from the position of the input device to the position of the robot (P2P), is good for precise position control [26]. This thesis focuses on proposing a novel 3D haptic trackball interface with P2P mapping for controlling tendon-driven continuum robots. This approach, compared with other interfaces, is accurate in point reaching and capable of haptic force feedback, which can help the user follow targeted paths and avoid obstacles near the robot tip.

1.1 Continuum robots

Early prototypes of continuum robots trace back to the last century, when people started to realize the limits of rigid link robots [28]. Over the past decades, the studies of continuum robots have increased significantly, including studies of bio-inspired continuum robots [17] [38], kinematics and dynamic modelling [16] [12], control methods [10] [21], sensing and estimation [3] [15], etc. Continuum robots are playing important roles in the fields of manipulation/grasping tasks and environment navigation, due to the advantages of their unique structure.

While traditional robots usually feature a structure similar with human arms or legs, which consist of several rigid arms with a small number of DOF, continuum robot are biologically inspired, mimicking the structures of snakes, elephant trunks and cephalopods, which consist of a high number of DOF. Their backbone-less structures enable them to navigate in constrained and complex environments, which are inaccessible for traditional rigid link robots [25]. A robot is said to be redundant when it has more DOFs than needed to finish a task. In extreme cases, where the robot has an infinite number of joints and zero link length, the robot becomes a continuum robot (also called hyper-redundant under this extreme case)[28]. Therefore, a continuum robot has an structure defined by an infinite-DOF elastic member [18].

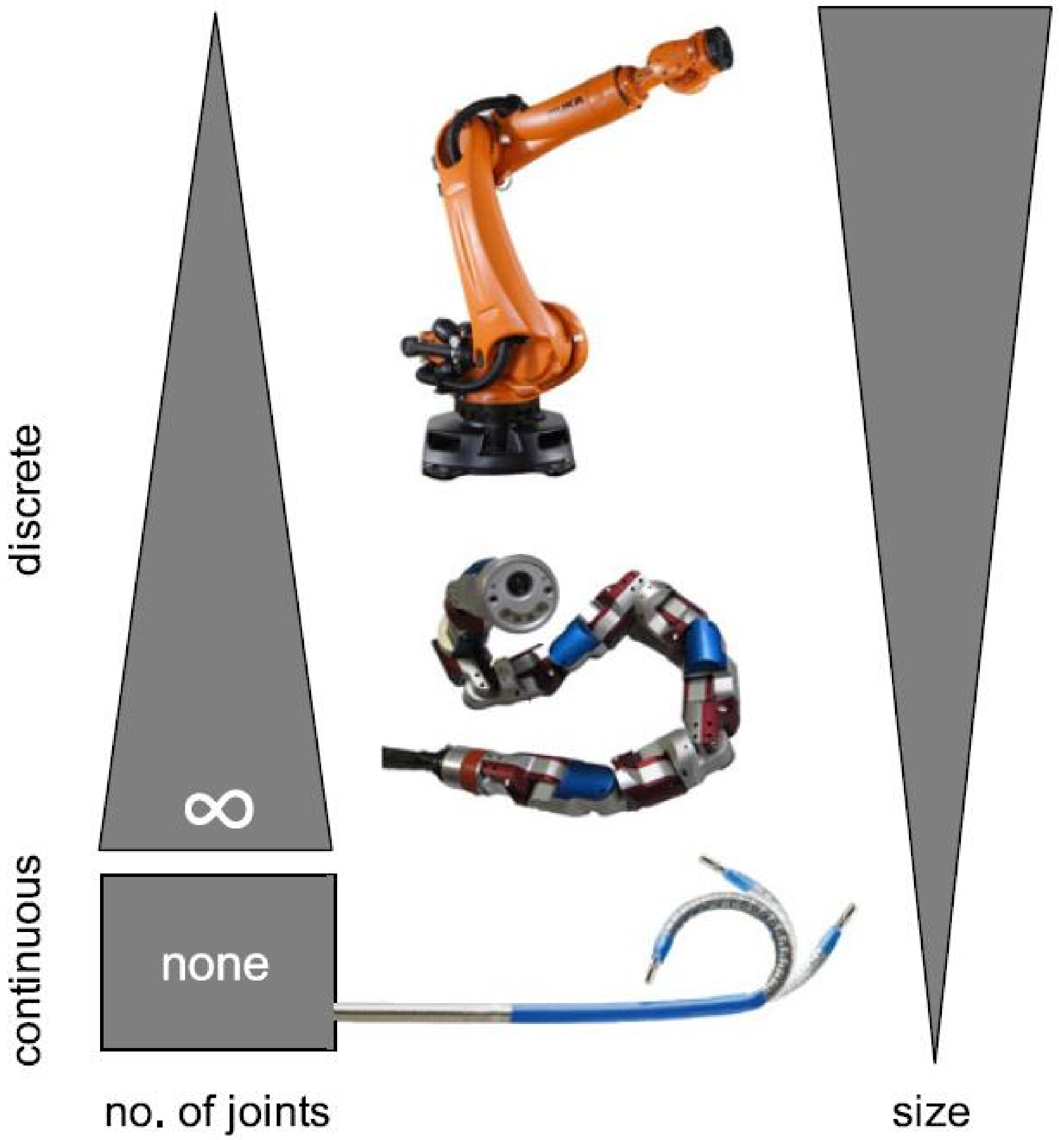

Figure 1.1. This figure, originally appeared in Jessica Burgner-Kahrs et al.[18], shows the number of DOFs from traditional rigid links robots to continuum robots(Images: top copyright from 2015 Kuka Robotics Corp.;bottom copyright from 2015 Hansen Medical Inc.)

Due to the flexibility and dexterity to work in constrained environments and ability to interact safely with the surroundings, continuum robots have been widely used in surgical applications, such as neurosurgery [34] [19] [36] and cardiovascular surgery [30]. Although biological structures seem simple, the way to recreate the beauty of nature is full of challenges, especially for surgical applications where a small error may result in severe injuries to organs. To overcome the difficulties of bringing robots into real surgeries, significant effort has been put into solving problems involving control, sensing and actuation [17].

1.2 Tendon-driven robots

There are two main categories of continuum robots, distinguished by their method of actuation. Intrinsic actuation usually features pneumatic or hydraulic chambers, and inflating or deflating chambers cause the deformation of the robot. However, due to the physical structure of chambers, the range of length per section is limited, because chambers have a certain physical length and it is usually not small. To realize a wide range of bending radii, a wide range of section lengths is preferred, and a zero section length would be ideal in extreme cases [2]. Therefore, extrinsic actuation, which allows for small diameter-to-length ratios is a better choice for surgical robots. For example, concentric tube robots, composed of several telescoping precurved tubes [27], and tendon-driven robots, made from several disks on a flexible backbone with pre-tensioned tendons, are widely used in minimally invasive surgery.

Compared with tendon-driven robots, concentric tube robots have a limited range of curvature based on the precurvatures of the tubes. Tendon-driven robots can reach a larger range of curvature by manipulating the length of each tendon. Yet, tendon-driven robots are not as easily miniaturized. While concentric tube robots have diameters below 2 mm, the diameter of tendon-driven robots tends to be larger than 3 mm [2] [19].

Like other continuum robots, the modelling of tendon-driven robot obeys an assumption of constant curvature, which means that each mutually tangent curved segment has a constant curvature along its arc length. In each segment, we use arc parameters, including curvature, length and twist angle, to describe the shape of the robot. Therefore, the position and orientation of every point on the robot can be derived from the arc parameters along each segment [18]. There are also studies aiming to develop models based on variable curvature, which consider a material-attached reference frame. However, these models generally require numerical integration to attain the closed-form solution of shape parameters [18] [31].

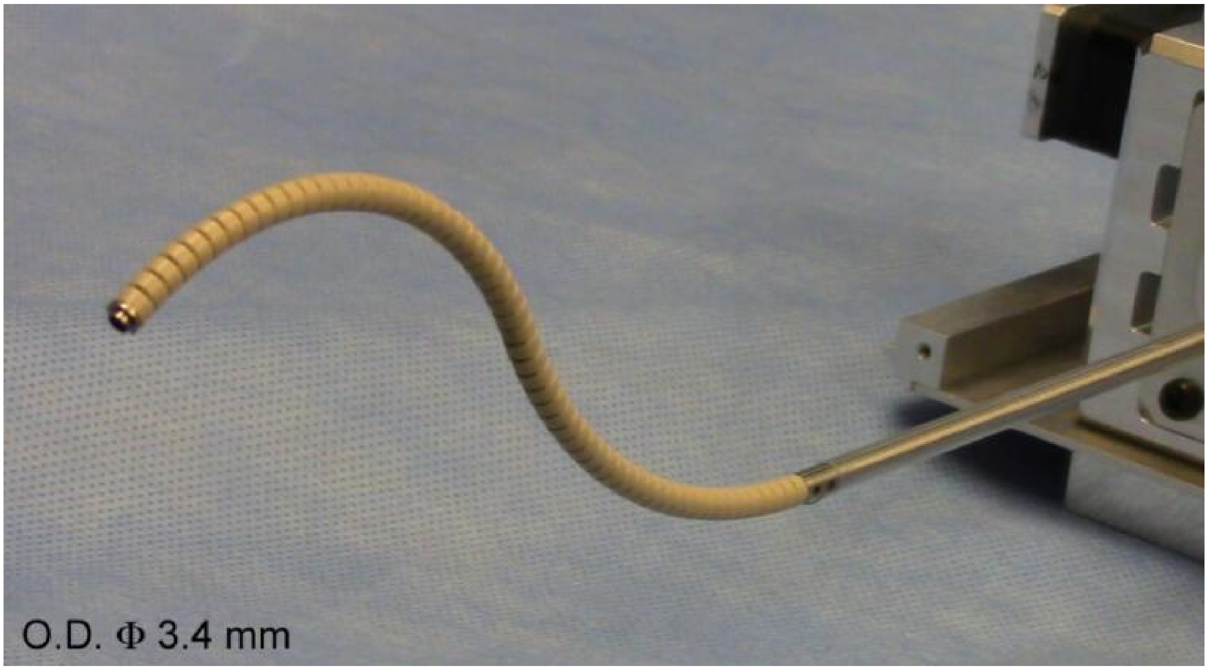

Figure 1.2. This figure, originally appeared in Takahisa Kato et al.[19], shows a tendon-driven robot with a diameter of 3.4mm

In this thesis, considering our preference of larger curvatures and the level of study, we choose to focus on applications of tendon-driven robots based on constant curvature assumption.

1.3 Conventional interfaces of surgical robots

There have been numerous studies on control systems, kinematics models, and newly-designed structures of continuum robots. However, research on the design of user interfaces for continuum robots are limited. Since continuum robots are designed to work in confined places, where humans have difficulties reaching, the robots are always teleoperated [29]. In teleoperation, the input device reading inputs from the human is called the “leader”, and the robot being controlled is called the “follower”.

In many studies, off-the-shelf joysticks have been a good choice for the leader device.

Joysticks are common, portable, and simple to use [22]. Also, joysticks usually come with several built-in buttons, which can be used to control different features according to the specific continuum robot follower.

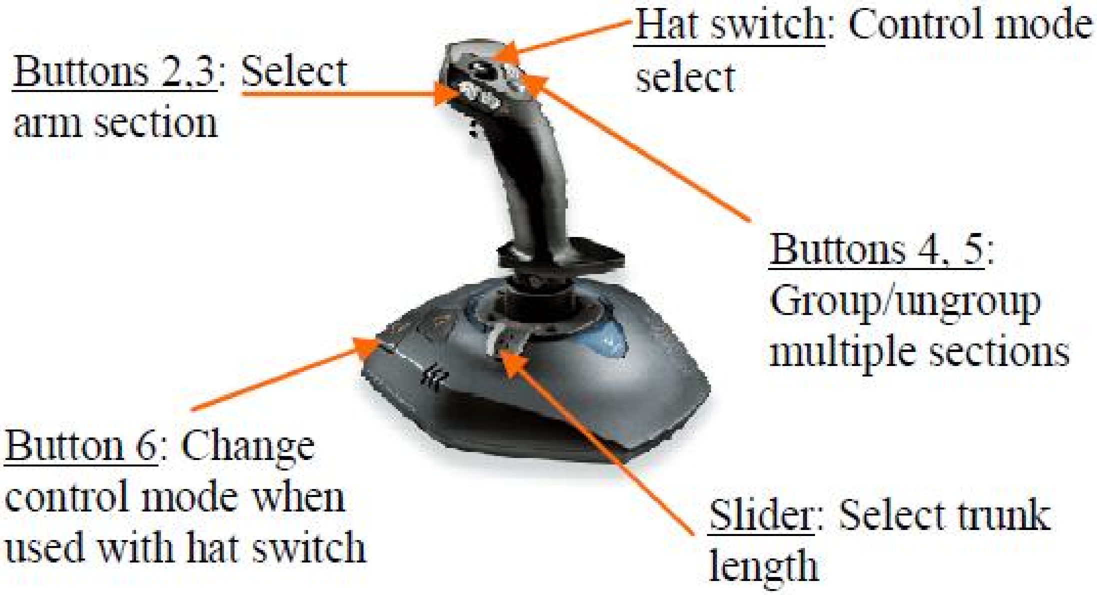
Hat switch: Control mode select
Buttons 2,3: Select arm section
Buttons 4, 5: Group/ungroup multiple sections
Button 6: Change control mode when used with hat switch
Slider: Select trunk length

Figure 1.3. This figure, originally appeared in Matthew Csencsits et al.[22], shows a joystick interface for controlling an Air-Octor continuum robot arm. Besides the mapping from the motion of joystick arm to robot tip position, there are several switches for changing control modes.

Recently there have been a few examples of designs that represent more intuitive interfaces and have a structure similar to that of a continuum robot [37] [14]. These interfaces involve designing a device that has a similar kinematic structure with the follower robot, and letting the follower continuum robot mimic the leader interface. The joystick works well when only controlling the tip position, but becomes non-intuitive and inappropriate when controlling the whole shape of the continuum robot [37]. Therefore, researchers have begun exploring interfaces with these more intuitive designs.

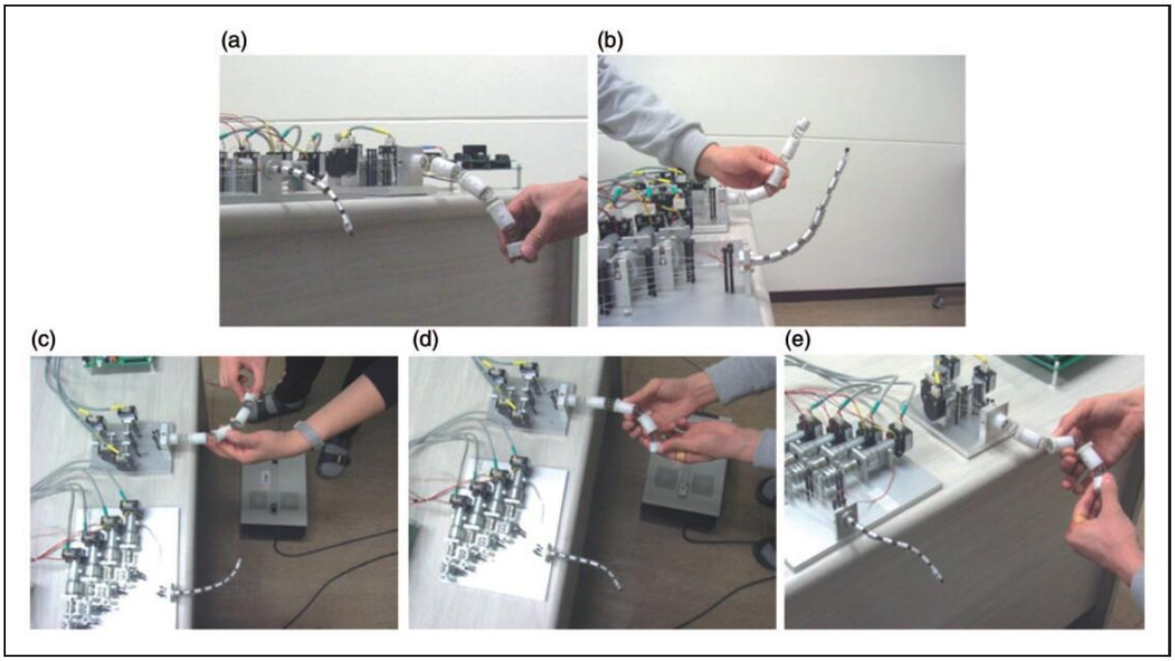

Figure 1.4. This figure, originally appeared in Hyun-Soo Yoon et al.[37], shows an intuitive interface for controlling a multi-moduled continuum robot. The interface, which is held by human hands, can not only control the tip position but also the body shape of the robot.

These interfaces perform well during experimental tasks, howver, they lack the ability to provide feedback to users. In confined workspaces, it is difficult to teleoperate the robot safely with high accuracy even for professionally trained operators. Especially in surgical tasks, the consequence of moving the robot in the wrong direction or accidentally touching any of the wrong sites by mistake, can be severe. Therefore, it is necessary to implement feedback to the operator in realtime, so as to improve the performance in practical scenarios.

1.4 Haptic interfaces

Haptic interfaces are designed such that ideally “what you see is what you feel” (“WYSIWYF”), which means the visual sensation and haptic sensation are consistent[35]. Earlier haptic systems were built to interact with virtual environments. The first haptic system, GROPE-1, displaying force fields for interacting protein molecules, was designed in 1967 [13]. Later, people started to use haptic interfaces as teleoperators as well.

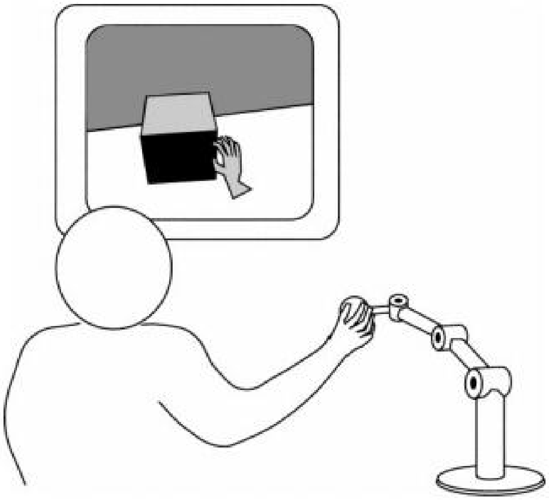

Figure 1.5. This figure, originally appeared in Yasuyoshi Yokokohji et al.[35], shows a typical haptic interface interacting with virtual environments.

Among the five senses, the sense of touch is the only one that provides a two-way interaction with the environment. The user can influence the environment by touching it via a robot, as well as feel the resulting effect via force feedback [23]. This feedback is highly useful for teleoperation. As a result, haptic interfaces have great potential as leader devices for robots, however, few studies have been focused on designing haptic interfaces for continuum robots. There have been some studies that use existing haptic interfaces to control a continuum robot [29] [6] [4]. The Phantom Omni, for example, is an electromechanical device with active force feedback in the x,y, and z axes, and is used most frequently in these studies [1].

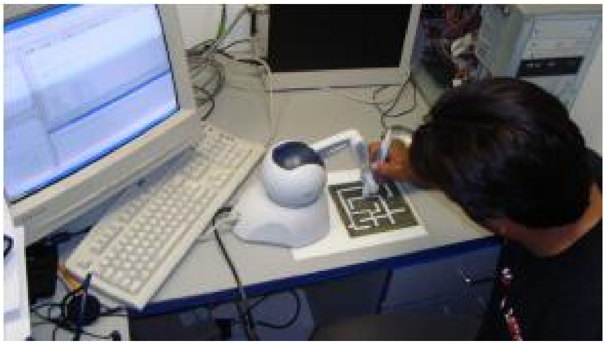

Figure 1.6. This figure, originally appeared in Bo Ouyang et al.[4], shows a Phantom Omni used as a input device.

The mapping used with the Phantom Omni, as well as with other similar haptic devices, is typically position to position (P2P) control, which maps the position of leader (input device) to the position of follower (robot). In contrast, joysticks typically map the position of the leader to the velocity of the follower. P2V devices are suitable for robots moving with high speeds in large workspaces, while P2P devices are more accurate in position control. Overall, P2V devices are more time efficient, while P2P devices have higher positional accuracy [26]. While the Phantom Omni has great positional accuracy, it has a limited workspace, naturally determined by its physical structure. An ideal design of haptic interface would be as accurate in position control as P2P devices, and would work as well in large workspaces as P2V devices.

1.5 Contributions

The contributions of this thesis are as follows. (1) We design a novel 3D haptic trackball interface. The interface is a leader device for teleoperating continuum robots. It can control the position of the robot tip in 3D space and provide force feedback to users when necessary, for example, when the robot is near an obstacle or deviating from a target path. The mapping scheme is P2P and is also capable of control in large workspaces, similar to P2V schemes. (2) We evaluate the interface by conducting a user study and comparing the performance of our haptic interface with a conventional joystick interface. The study contains two tasks —

following paths and avoiding obstacles. The results of the study prove that our interface has better performance than conventional joystick interfaces in terms of precision and obstacle avoidance.

Chapter 2

System Design

Our 3D haptic interface can be divided into two parts, a 2 DOFs traditional 2D trackball and a 1 DOF linear capstan. The traditional 2D trackball tracks the input and provides feedback on x and y axis. The linear capstan drive is in charge of motions on z axis. The motors are controlled by an Arduino Mega board. The interface is served to control the robot tip position in 3D space.

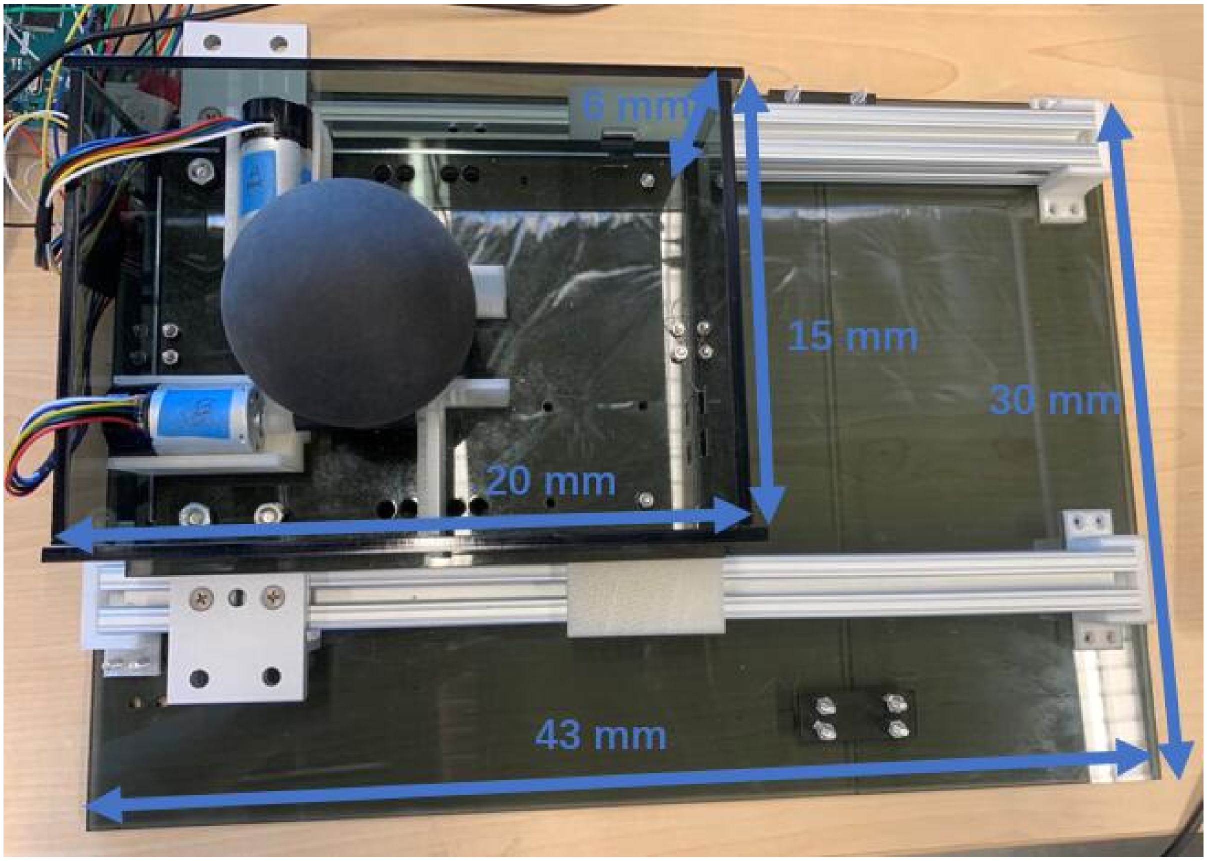

Figure 2.1. An overview of our 3D haptic trackball interface with dimensions.

2.1 Hardware design

The 3D trackball is an updated version of traditional 2D trackballs. Trackball interfaces usually have a friction-drive mechanism, which means the rotation of the trackball will drive a shaft by friction, and the rotation of the shaft will be further transferred to motors, as shown in previous studies [24] [20].

Our design uses a similar friction-drive mechanism for the 2D trackball portion, and implements a linear capstan-drive under the 2D trackball, which controls the motions in the third dimension. There are two 12V HP motors with 48 CPR encoders from Pololu for the 2D part of the system, each extended by a motor shaft. Two more shafts are used for support, and they are not connected to any motor and can thus spin freely. Shafts and other supporting parts are 3D-printed. Every shaft is wrapped with neoprene tubing to increase the friction between the shaft and the trackball. We use a polyurethane rubber ball as our trackball. The diameter of shafts with neoprene is 8 mm, and the diameter of the trackball is 28 mm. The motors have built-in encoders, which can measure the amount and direction of rotation while the motors are spinning.

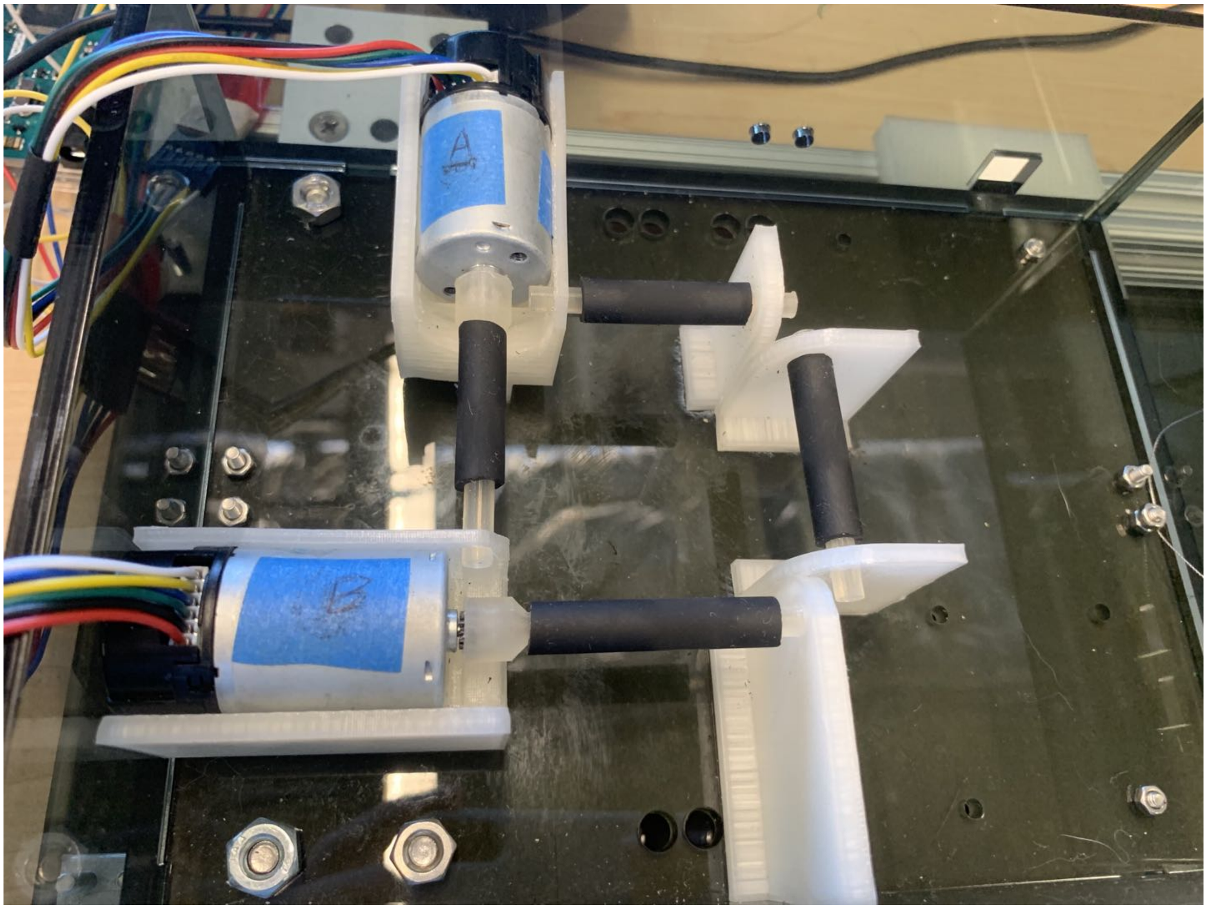

Figure 2.2. Motors and shafts in the 2D part of the system.

The linear capstan-drive is a mechanism to transfer forces between tensions in the tendons and torque output on the motor shaft.. There is another 12V HP motors with 48 CPR encoders under the 2D trackball part, connected with a linear capstan drive. A long tendon (an extra-flexible weather-resistant coated wire rope) is wrapped over the motor shaft, with the two ends fixed on the box containing the 2D trackball part. When the user moves the box along the aluminum track, the third motor underneath will be driven by the linear capstan-drive, and the encoder in the motor can read the length and direction of the movements. Similarly, force feedback can be applied through the inverse way, when the motor is actively blocked (or move in the opposite direction of current velocity) and the user can no longer move in the original direction, as shown later in figure 2.6.

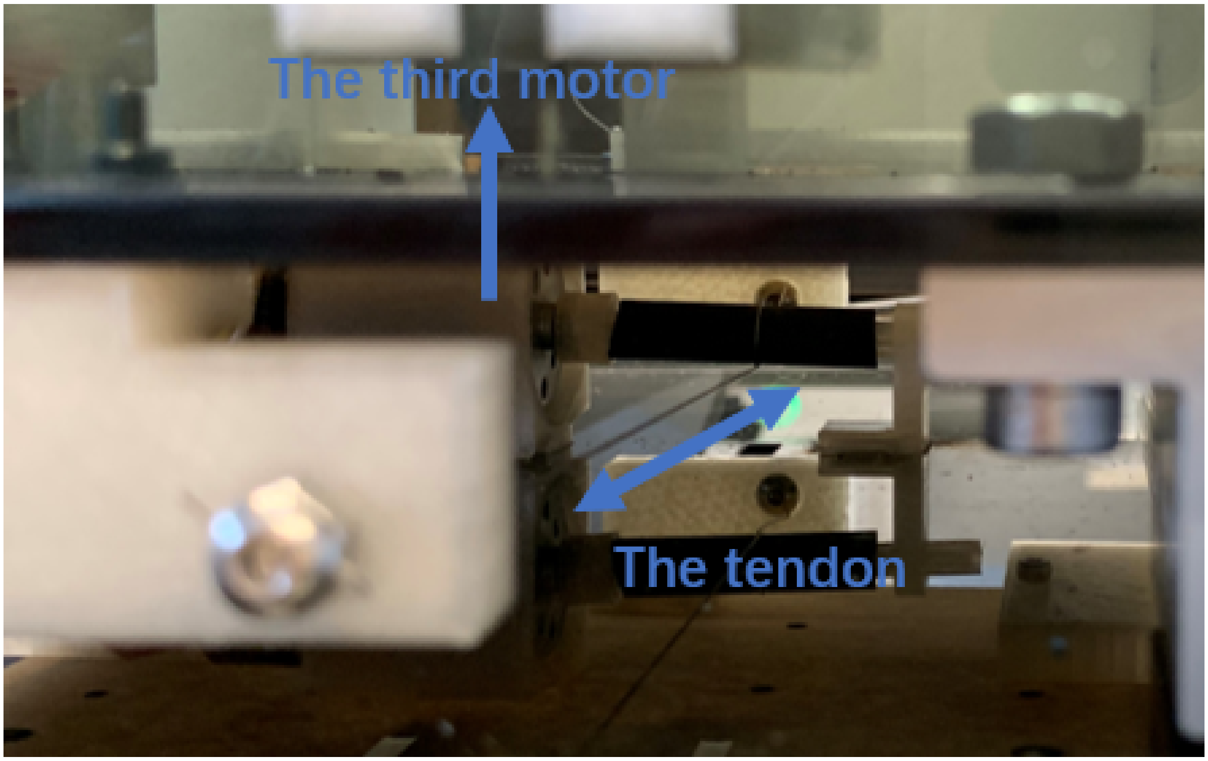

Figure 2.3. The linear capstan-drive connected with the third motor.

With the linear capstan-drive, our trackball interface is capable of sending inputs and providing force feedback in 3D space.

2.2 Haptic feedback

Force rendering is important in haptic systems. In virtual environments, the simplest force that we can render is the resisting force from a virtual wall. The magnitude of this kind of force comes from the spring model. Imagine we are controlling a virtual proxy, and when the proxy is in contact with the virtual wall (or penetrating the virtual wall), we can calculate the haptic force feedback as:

$$F = k(x_{wall} - x_{user}), \tag{2.1}$$

where k is the predefined stiffness of the virtual wall. In applications of teleoperation, things are similar, except that we are controlling robots in real environments.

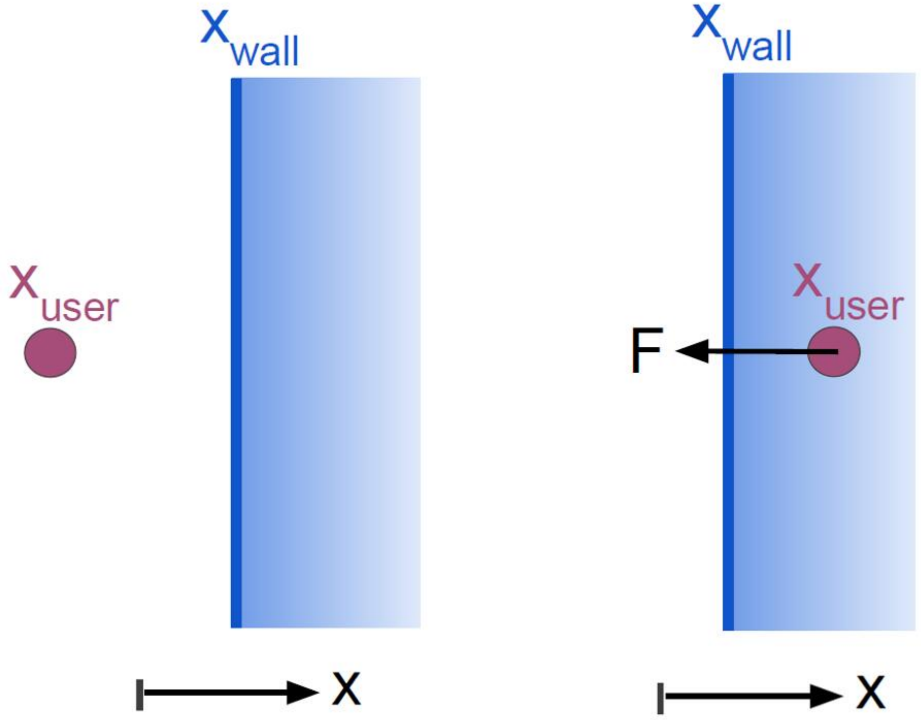

Figure 2.4. This figure shows the haptic feedback of a virtual wall.

In our 3D haptic trackball system, there are two kinds of force feedback. The first one is force feedback on the trackball. Based on the spring model, we can calculate the magnitude of the force rendered by the motor connected with the trackball, also scaled by a predefined stiffness:

$$F = k \cdot \Delta x, \tag{2.2}$$

where k is defined to be 500 N/m and Δx is dependent on the task we perform with the robot (see the chapter on the user study). The limit of force rendered is determined by the parameters of the motor. The motor we use has a maximum output torque of 0.2 $kg \cdot cm$. Assuming quasi-static conditions, the force transferred by friction-drive is equal on both sides, giving the following relationship:

$$F_{feedback} = F_{shaft} = \frac{\tau_{motor}}{r}, \tag{2.3}$$

where $r = 4$ mm is the radius of motor shaft and $R = 14$ mm is the radius of trackball. Applying the maximum torque from the motor, gives the maximum force feedback, which is equal to 5 N.

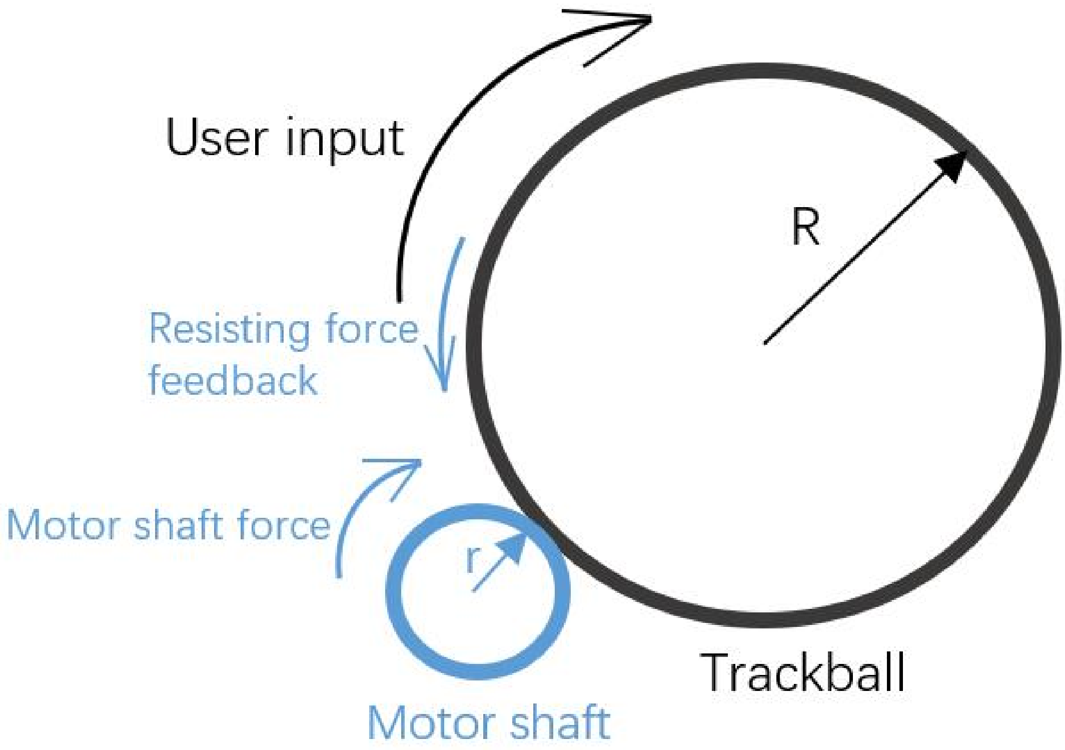

Figure 2.5. The mechanism of force feedback on the trackball for a single motor shaft. The other motor shaft is oriented perpendicularly to the motor shown in this figure.

The second kind of force feedback is force on the linear capstan-drive. Similarly, it is also friction that transfer the motor force to force feedback. A tendon is wrapped over the motor shaft without slipping. The force feedback can also be calculated as eq(2.3), thus the maximum force is 5 N as well.

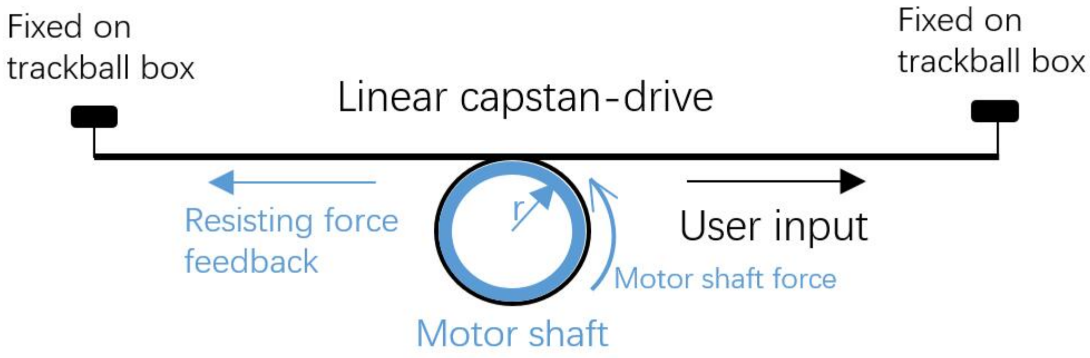

Figure 2.6. The mechanism of force feedback on the linear capstan-drive.

Chapter 3

Teleoperation Methods

We use our 3D haptic trackball to teleoperate a tendon-driven continuum robot. Traditional tendon-driven robots are usually only capable of motions with 2 DOFs in each section of constant curvature. To deploy it along a tortuous path, we need to add an additional translational DOF [2]. Our design includes a linear actuator, driven by a stepper motor, which is added to a standard tendon-driven robotic system. With the linear actuator, the tendon-driven robot can also move along the z-axis (defined as the direction of insertion along its backbone). This motion is teleoperated by the linear capstan-drive in our haptic interface.

The kinematics of tendon-driven robots are defined by arc parameters (κ, ϕ, l) in each section, and the tip position (x, y, z) can be calculated through the forward kinematics. Since we aim to control the tip position directly using our interface, we need to calculate the arc parameters using the input signal of tip position through the inverse-kinematics. Furthermore, since part of the motion on the z-axis is coupled with x and y by the nature of the tendon-driven robot, we need to compensate for such amount of coupled displacement on z-axis, which can also be calculated by the kinematics model.

To get the actual tip position, we need a tracking system that works in real time. We apply a vision tracking algorithm based on two webcams, which localizes the robot tip in 3D space. The code for robot controller, inverse-kinematics and vision tracking algorithm are written in MATLAB, while the functions of haptic interface are written in Arduino. To integrate functions and signals in the two different softwares, we use techniques of serial communications between MATLAB and Arduino.

3.1 Linear insertion

A linear actuator, driven by a Nema23 stepper motor, is implemented on a tendon-driven robot with 2 sections. The length of the first section is 250mm and the second section is 240mm. There are 6 tendons in the first section, oriented 60° apart. Three of them extends to the end of the second section, with 120° apart. Therefore, this tendon-driven robot has 2 independent sections with constant curvature, resulting in 4 DOFs.

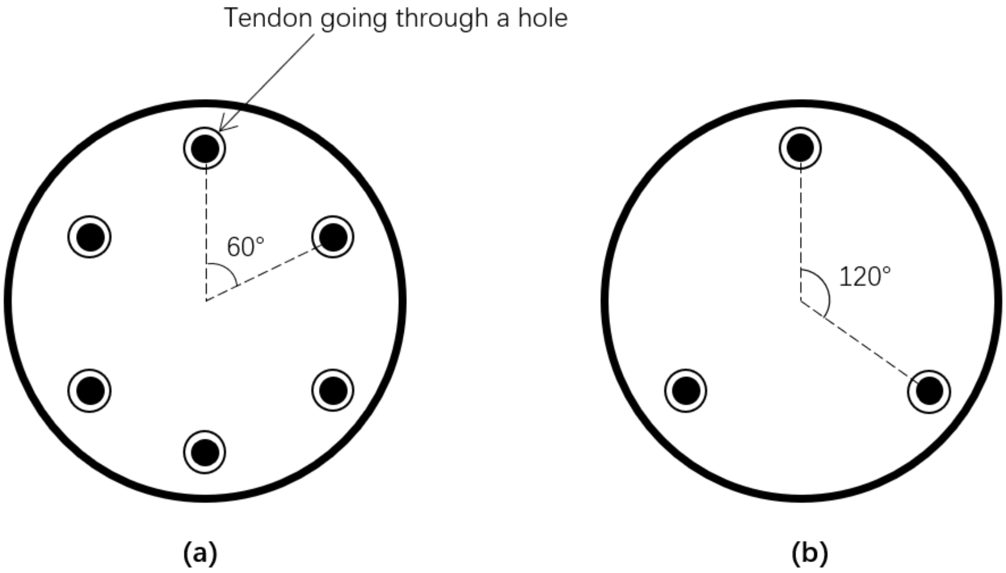

Figure 3.1. The layout of tendons going through spacer disks in two sections: (a) the first section with 6 tendons, (b) the second section with 3 tendons.

We fix the robot, as well as 6 motors, each controlling one tendon, on an aluminum track, then fix the aluminum track on the slide table of the linear actuator. We use TB6600 motor driver to drive the stepper motor, controlled by an Arduino Mega board. There is no encoder for the stepper motor, since stepper motors are controlled in open-loop. One encoder count from the

DC motor of the haptic interface is programmed to move 50 steps of the stepper motor, which results in 0.3 mm movement of the slide table. The total travel length of the haptic interface along the z-axis is 150 mm, corresponding to 300 encoder counts of the DC motor. Therefore, the mapping from leader to follower is computed as:

$$\frac{z_{leader}}{z_{follower}} = \frac{\frac{150 \text{ mm}}{300}}{0.3 \text{ mm}} = 1.67 \tag{3.1}$$

The following figures show tendon-driven robot from different views.

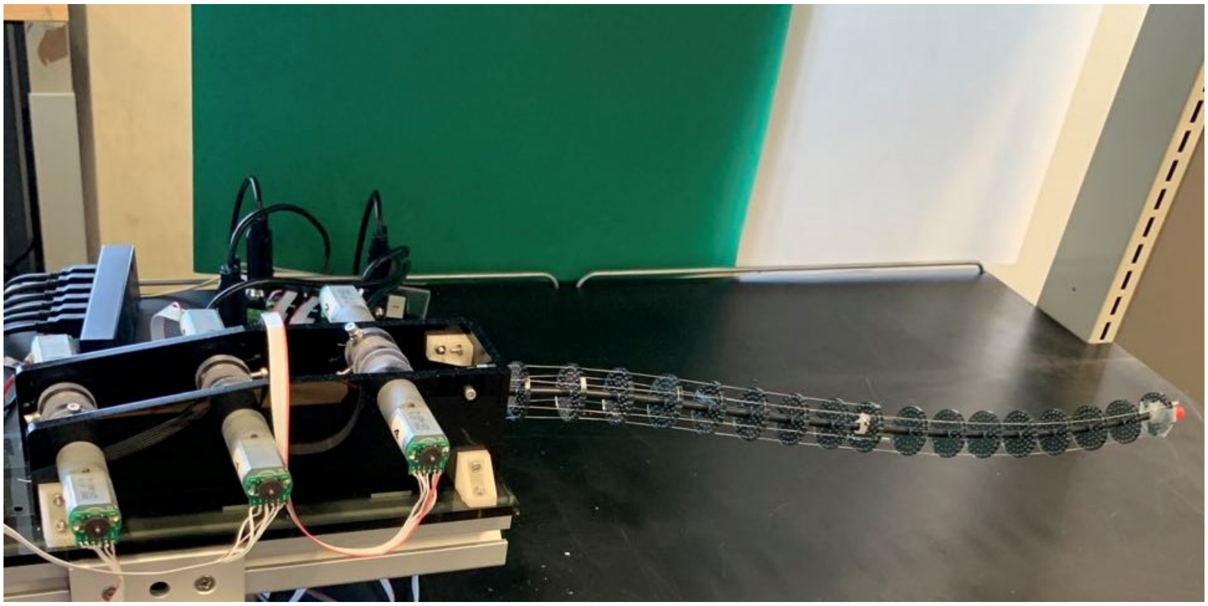

Figure 3.2. The backbone and driving components of the tendon-driven robot.

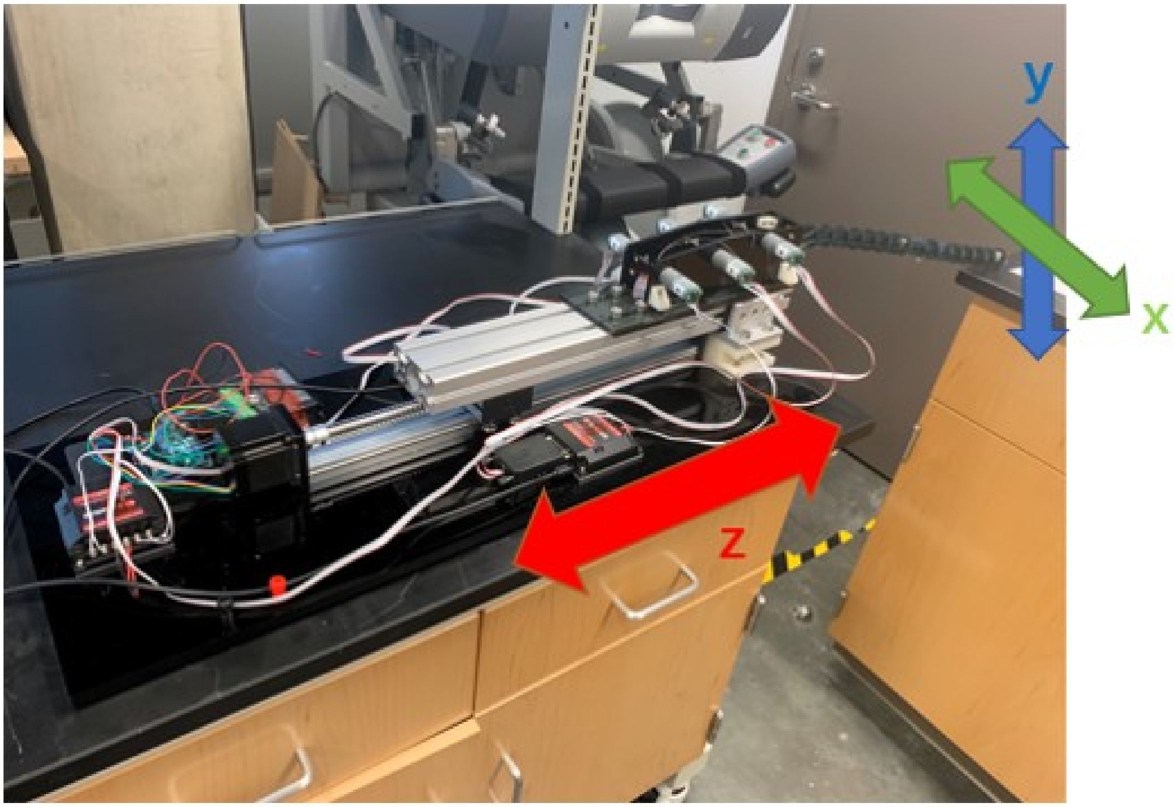

Figure 3.3. The tendon-driven robot with linear actuator as a whole system.

3.2 Inverse kinematics

The parameters of continuum robots are defined in three spaces: actuator space, configuration space and task space. The mapping between actuator space and configuration space is called the robot-specific mapping, since the actuation of different continuum robots varies, which results in different parameters in actuator space, thus the mapping is dependent on the type of robot. For example, tendon-driven robots are defined by tendon length in actuator space. The mapping between configuration space and task space is called the robot-independent space, since all the continuum robots are defined by arc-parameters (κ, ϕ, l), and parameters in task space are always the position and orientation of robot in physical space, thus the mapping is universal for continuum robots [33].

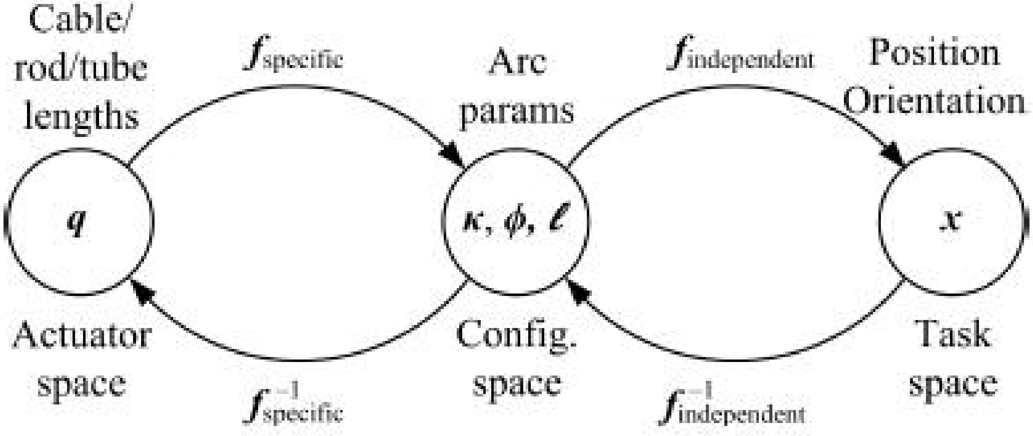

Figure 3.4. This figure, originally appeared in Robert J. Webster III et al. [33], shows the mappings in continuum robots between actuator space, configuration space and task space.

The robot-specific mapping is described in [33], where equations for deriving tendon lengths from arc parameters are clearly shown, and it is used to control the original tendon-driven robot with 2 DOFs by default. Since we are using parameters in task space as inputs, we need to do the inverse kinematics from task space to configuration space, which is $f^{-1}_{independent}$. This mapping, however, is more challenging and not as straightforward. We decide to use a Jacobian-based method combined with Newton-Raphson method to solve the inverse kinematics iteratively.

Continuum robots can consist of several sections, each having a constant curvature. Each section can be considered as an arc, defined by arc parameters. This assumption is made for all the continuum robots. In practice, however, due to the influence of manufacturing errors, external loads and interactions with environments, the constant curvature assumption may not be valid in some cases. Considering that in our study we don't include external load and interactions with environments are limited, we hold this assumption in our kinematics analysis. In addition, we will implement tip tracking method for obtaining actual tip position. For our kinematic model, we can first consider the simplest case, a single section model, then build the remaining sections based on the tip position of the first section. The single section model is shown in the following figure:

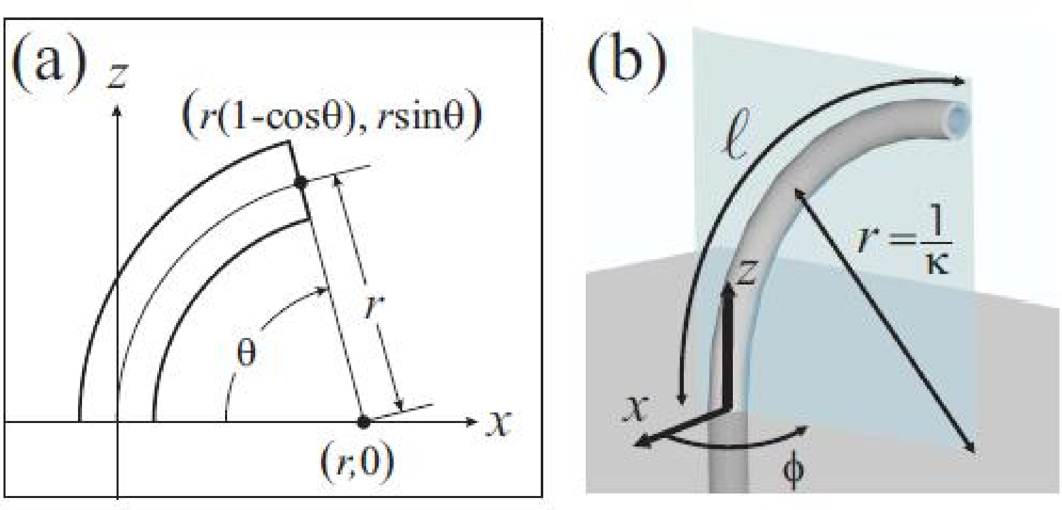

Figure 3.5. This figure, originally appeared in Robert J. Webster III et al. [33], shows the single section model of a continuum robot. (a) When ϕ is zero, the robot is in the x-z plane. (b) As ϕ changes, the arc rotates along z-axis.

Here we define the variables that will be used in the derivation:

Table 3.1. Variables used for inverse kinematics.

Variable symbol	Variable name
T	Transformation matrix
$R_{(\theta)}$	Rotation matrix of a space angle θ
p	Position vector in transformation matrix
θ	Arc angle
ϕ	Rotation angle of the arc (one of the arc parameters)
κ	Arc curvature (one of the arc parameters)
l	Arc length(one of the arc parameters)
r	Arc radius
x_{tip}	tip position
x	x component of tip position
y	y component of tip position
z	z component of tip position

The forward kinematics are:

$$T_1 = \begin{bmatrix} R_{(\theta_1)} & p \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_{(\phi_1)} & 0 \\ 0 & 1 \end{bmatrix} \tag{3.2}$$

where

$$R_{(\theta_1)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\theta_1 & sin\theta_1 \\ 0 & -sin\theta_1 & cos\theta_1 \end{bmatrix} \tag{3.3}$$

$$R_{(\phi_1)} = \begin{bmatrix} cos\phi_1 & -sin\phi_1 & 0 \\ sin\phi_1 & cos\phi_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.4}$$

$$p = [r(1 - cos\theta), 0, rsin\theta]' \tag{3.5}$$

$$r = \frac{1}{\kappa} \tag{3.6}$$

$$\theta = \kappa l \tag{3.7}$$

Since l is a fixed length and κ can be expressed in terms of θ and l, we can express everything in terms of two angles (ϕ, θ). The tip position $x_{tip} = (x, y, z)'$ is:

$$T_1 = \begin{bmatrix} R & x_{tip} \\ \mathbf{0} & 1 \end{bmatrix}$$
$$= \begin{bmatrix} cos^2\phi_1(cos\theta_1 - 1) + 1 & sin\phi_1 cos\phi_1(cos\theta_1 - 1) & cos\phi_1 sin\theta_1 & cos\phi_1 \frac{l}{\theta_1}(1 - cos\theta_1) \\ sin\phi_1 cos\phi_1(cos\theta_1 - 1) & cos^2\phi_1(1 - cos\theta_1) + cos\theta_1 & sin\phi_1 sin\theta_1 & sin\phi_1 \frac{l}{\theta_1}(1 - cos\theta_1) \\ -cos\phi_1 sin\theta_1 & -sin\phi_1 sin\theta_1 & cos\theta_1 & sin\theta_1 \frac{l}{\theta_1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.8}$$

For N sections, the tip position is:

$$\begin{bmatrix} R & x_{tip} \\ \mathbf{0} & 1 \end{bmatrix} = \prod_{i=1}^{n} T_i \tag{3.9}$$

The tendon-driven robot we use has 2 sections, so the transformation matrix $T_1 T_2$ can be used for tip position forward kinematics.

The inverse kinematics involves solving for $(\phi_1, \phi_2, \theta_1, \theta_2)$ using

$$(T_1(\phi_1, \theta_1)T_2(\phi_2, \theta_2))[:, 4] = [x, y, z, 1]' \tag{3.10}$$

where T[:,4] represents the 4th column of matrix T. Equivalently, turning equation 3.10 into the form of function in Newton-Raphson method, we solve the function:

$$F = (T_1(\phi_1, \theta_1)T_2(\phi_2, \theta_2))[1:2, 4] - [x, y]' = 0 \tag{3.11}$$

For simplicity in calculation, we only take the x and y component from the function to solve since the position on z-axis is coupled with positions on x-axis and y-axis for the tendon-driven robot.

With Newton-Raphson method, starting from an initial guess of $\alpha_0 = (\phi_{1_0}, \phi_{2_0}, \theta_{1_0}, \theta_{2_0})$, the k-th iteration will be:

$$\alpha_{k+1} = \alpha_k - J_k^{-1} F_k \tag{3.12}$$

where J_k is the jacobian of F_k on $\alpha_k = (\phi_{1_k}, \phi_{2_k}, \theta_{1_k}, \theta_{2_k})$. Since J here is non-invertible (F is not a square matrix), we should use the pseudo inverse J_k^+:

$$J_k^+ = J_k^T (J_k J_k^T)^{-1} \tag{3.13}$$

When F is small enough (we use $|F| < 0.01$ as the condition), we get the resulted arc parameters $(\phi_1, \phi_2, \theta_1, \theta_2)$ from Newton-Raphson method.

Once we solve the inverse kinematics, we are able to control the robot in x-y plane by inputs from the 2D trackball. We set the scaling factor of mapping as:

$$\frac{x_{leader}}{x_{follower}} = \frac{360^\circ}{425 \text{ mm}} \tag{3.14}$$

The mapping for y-position is the same.

3.3 Mappings in 3D space

Section 3.1 explains the mapping of the z-axis, and section 3.2 explains the mapping in x-y plane. The mapping in 3D space, however, is not simply the combination of these two. Since the tip motion in x-y plane will also generate a coupled motion on z axis, we need to compensate for this coupled motion, so that the teleoperation will, for example, only result in motion along the x-axis if the input is (x,0,0).

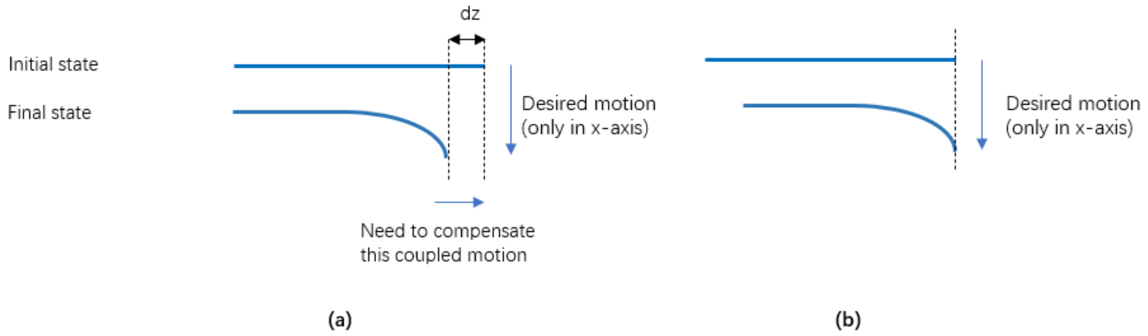

Figure 3.6. Compensation of coupled motion on z axis: (a) Without compensation, (b) With compensation

The new z-axis position in the tendon-driven robot model can always be computed from the forward kinematics, as in equation (3.8). Then we have the coupled displacement:

$$z_{coupled} = z_{FK,new} - z_{FK,old} \tag{3.15}$$

where z_{FK} is the z-position computed from forward kinematics, and the terms ”new” and ”old” refer to the states before and after a control action is executed.

Subtracting the coupled displacement, we have the final displacement on z axis:

$$z = z_{follower} - z_{coupled} \tag{3.16}$$

3.4 Camera-based vision tracking

The estimated tip position from the kinematics model is not completely accurate, since the tendon-driven robot will always bend downward slightly due to the gravity of its own weight, and the errors of motors and encoders can be another factor here. To get the actual tip position, we use a camera-based vision tracking system. For the cameras to recognize the robot tip, we apply a color-based algorithm and attach a small sphere with a distinct color from the background on the robot tip.

The reason for using a small sphere to label the robot tip is because the sphere will always remain the same shape despite the viewing angle between the robot tip and camera lens, so that we can always use the mass center of the sphere to represent the exact tip position. Meanwhile, the material should be lightweight, so that it will not aggravate the bending issue. We 3D print a sphere with a diameter of 4 mm attached with a thin cylinder for connection. Then we paint the sphere in red to distinguish it from the background.

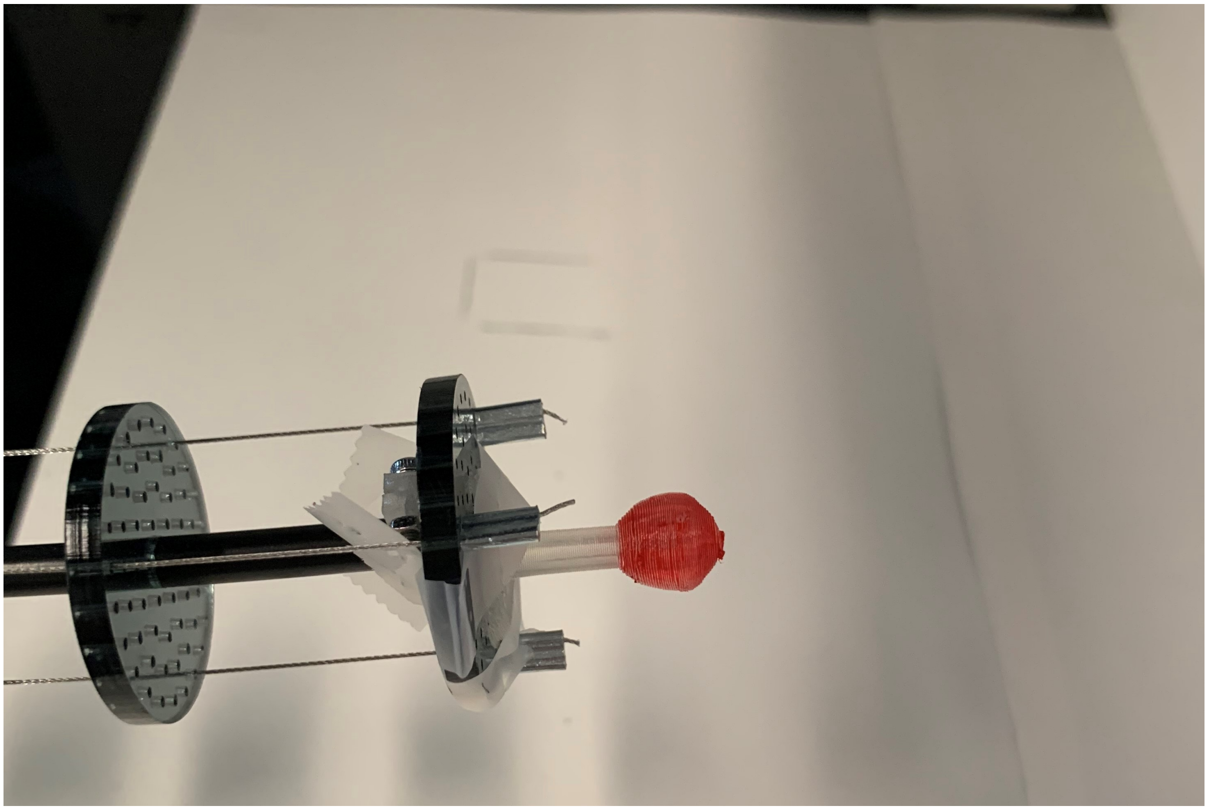

Figure 3.7. The 3D printed sphere painted in red for color tracking.

The color-based algorithm is basically a filter for red color. The 60 fps camera will take a screenshot 60 times per second. For each screenshot, we first obtain all the red components passing through a RGB threshold we set in the image, then convert the whole image into binary image and find the bounding box and centroid of red regions using blob analysis. Blob analysis is a method for recognizing regions with similar patterns (color, shape, etc.), and it is a built-in function in MATLAB. After finding all the red regions, we need to further filter out small regions that are selected by mistake due to reflection or other influence of natural light. This is simply done by selecting the region with the largest area, since the disturbing regions are always very small (most likely small dots).

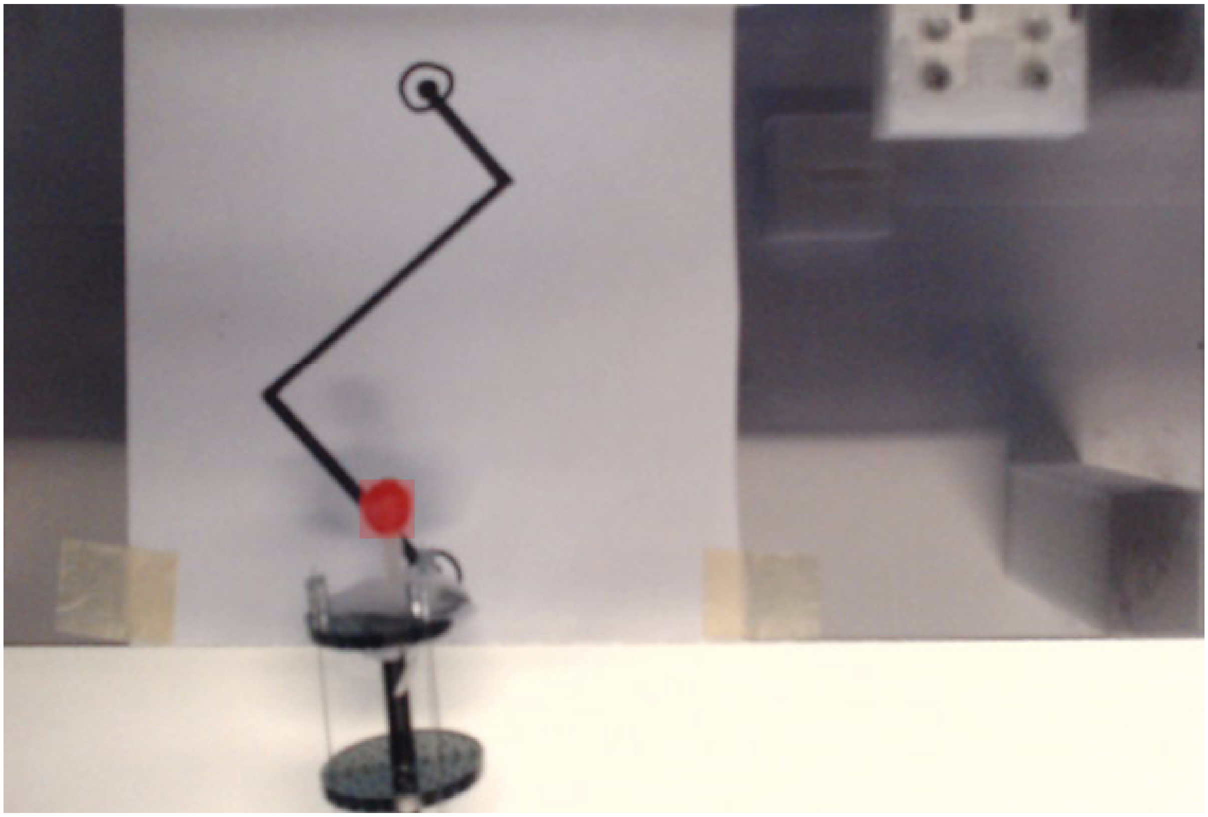

Figure 3.8. The red tip in camera view and its bounding box recognized by blob analysis.

We use 2 cameras for tracking. Each camera is capable of vision tracking in a 2D plane, and the combination of 2 perpendicularly oriented cameras is capable of vision tracking in 3D space. The camera frame is a 640*480 pixel image for each camera. We calibrate the camera by measuring lengths of different objects in pixels and then converting it into millimeters. The mapping from camera frame to physical space is:

$$1 \text{ pixel} \sim 0.3 \text{ mm} \tag{3.17}$$

3.5 Software implementation

The program is mainly written using MATLAB. However, the 3 motors in the haptic trackball interface are controlled using Arduino. Since the motors in the interface need to transfer the user input to the robot, and the tracking system written in MATLAB needs to transfer back the actual tip position so that the interface can decide whether to provide force feedback as well as how large the force should be, we need to communicate between MATLAB and Arduino. We use serial port communication to realize this function. The overall workflow of the signal is shown in the following figure:

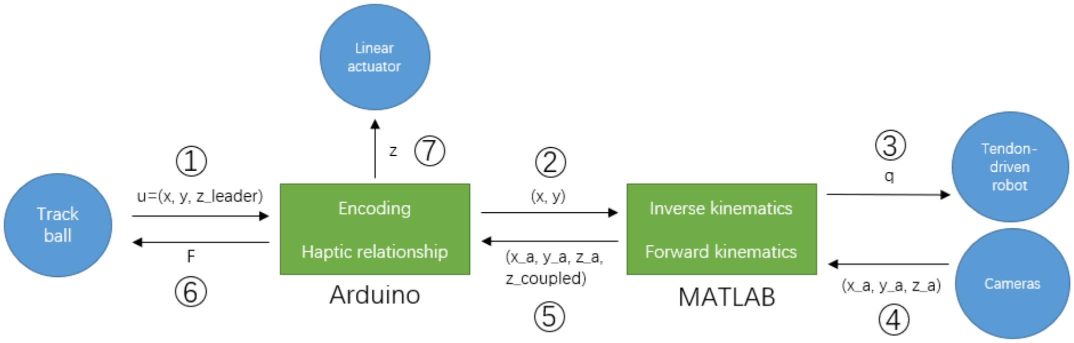

Figure 3.9. The workflow of the signals in the system.
1: User input: $u = (x, y, z_{leader})$.
2: Encoder reading (x, y).
3: Tendon lengths q.
4: Actual tip position (x_a, y_a, z_a).
5: Actual tip position (x_a, y_a, z_a) and coupled z-displacement $z_{coupled}$.
6: Force feedback F
7: Final z-displacement $z = z_{follower} - z_{coupled}$.

As shown in the figure, 1 is the user input, and the encoder read desired x y z position in Arduino. Then the encoder readings of x and y input are sent to MATLAB, and input z is saved for linear actuator. In 3, MATLAB send tendon lengths to the tendon-driven robot according to desired x y input, so the robot tip will move. As the robot moving, the camera near the robot send actual tip position back to MATLAB, and MATLAB further send this information plus the calculated coupled motion back to Arduino. Then in 6 Arduino will translate this information into force feedback provided by the trackball, at the meantime it will send the final z-displacement to linear actuator as in 7, so the robot will know how long it should move along z-axis. This whole loop is a complete control loop for one user input. The loop rate for this is around 15 Hz. The time consumed for each loop varies frequently in MATLAB, but stays within a certain range. The serial communication is an important factor that decreases the loop rate. Without serial communications, the loop rate can reach 30 Hz when we control this robot.

Chapter 4

User Study

We designed a user study to evaluate the performance of our proposed interface by comparing it with an off-the-shelf interface, a joystick on a gamepad. The study is done within a confined workspace, with 2 cameras mounted on the right wall and top roof for vision tracking. We setup the joystick according to the anticipated functions, and design a clutch button to better control the robot. All the mappings between the interfaces and the robot, as well as the orientation of the setup, are tested by several users in unofficial pilot tests prior to running the study. After the study, we ask users to fill out a post-study survey. All qualitative data from the survey as well as quantitative data recorded during the study are then analyzed.

4.1 Study design

4.1.1 Device setups

The study involves using two different interfaces to control a tendon-driven continuum robot. Users will be controlling this robot using two different interfaces. The first interface is our 3D trackball interface that can provide haptic feedback, and the other one is an off-the-shelf joystick interface on a gamepad. To further evaluate the effect of force feedback, we design a switch to control the presence of the force feedback on the 3D trackball interface. Users will be doing 2 different tasks, each using 3 setups of the interface: the joystick, the trackball with force feedback, the trackball without force feedback. To prevent the user from hearing the sound of motor, which will indicate the force feedback by sound instead of haptic feedback, users will wear a noise-cancelling headphone in trials which is possible to generate force feedback.

4.1.2 Training section

Before the tasks begin, we have a training section for users to get familiar with the devices. We will ask the user to control the robot by moving it in 6 basic directions (+x, -x, +y, -y, +z, -z) with two different interfaces. Once the users are clear about the mapping, they will control the robot in a simple virtual environment using the trackball interface to feel the force feedback, as shown in figure 4.1.

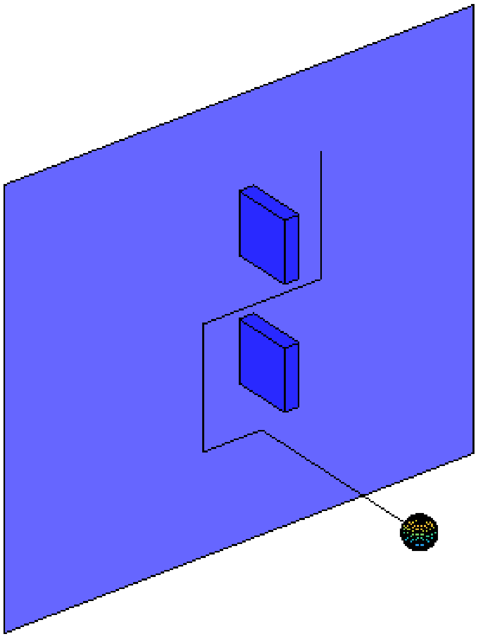

Figure 4.1. The virtual environment in MATLAB. The user will use the haptic trackball interface to control the sphere and hit the virtual walls to feel the force feedback.

4.1.3 Following paths task

The first task is following paths. In this task, users control the robot to follow two successive planar paths, using 3 different device setups, each with 3 trials (repeat the same task three times). The path contains two parts. The first part of the path is in the x-z plane. To avoid the influence of viewing angle, users will see the path from the top camera, which will be shown on the laptop screen (as shown in Fig.3.8). Once users reach the endpoint of the first path, they can start following the second path, which is on the x-y plane. Since users are directly facing this path, they will not need the camera view for this part. During the task, the actual trajectories of users will be automatically recorded for data analysis.

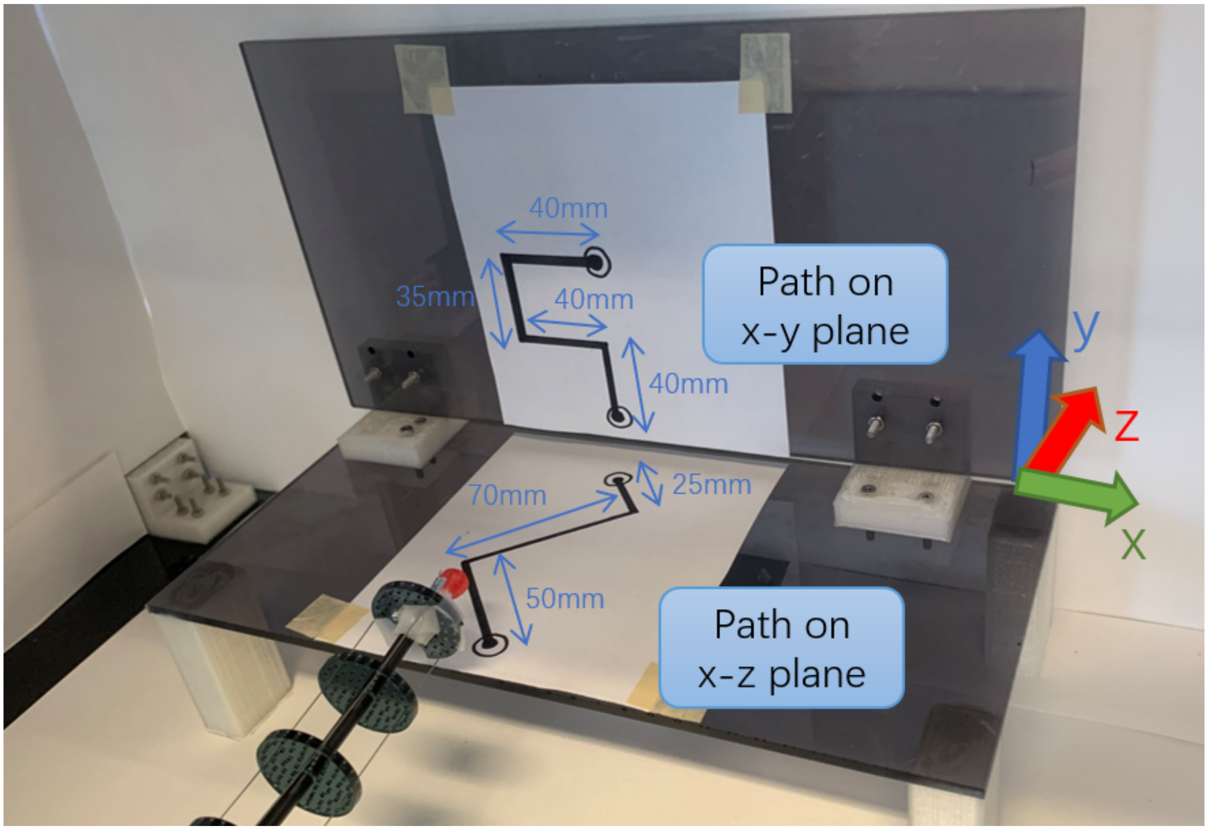

Figure 4.2. The layout of following path task.

The path is a curved line with a width of tolerance for force feedback. The width is set to be 5 mm on each side. Within the width, there will be no force feedback. The value of the width is tested to be noticeable for users to feel the force feedback, while not being too strict. Once the robot tip is out of the path, there will be force feedback to help users control the robot back to the track. Force feedback in this task is designed to be a spring force, as mentioned in previous chapters. The Δx here is the distance between the proxy position mapped from user input and the given path, decomposed into two normal directions (x_1, x_2). Force feedback is thus $F_1 = kx_1$, $F_2 = kx_2$, where k=500 N/m.

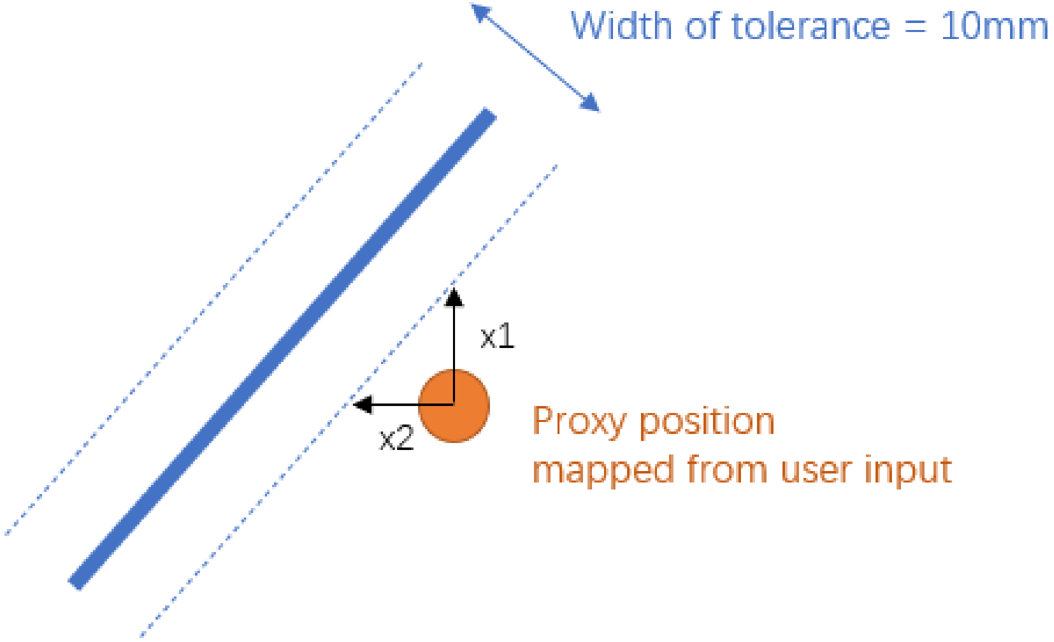

Figure 4.3. The force feedback in following paths task.

4.1.4 Avoiding obstacles task

The second task is avoiding obstacles. In this task, users will control the robot to move from a starting point to checkpoint one, then to checkpoint two, while avoiding the obstacles between them. The $30 \times 30 \times 30$mm clear boxes represent obstacles and the 10×10mm green squares are checkpoints. The workspace setup is shown in figure 4.4. Users will look directly to the work environment during the whole task. Similarly, users will use 3 different device setups, each with 3 trials. The number of times the robot tip hits the obstacles will be recorded.

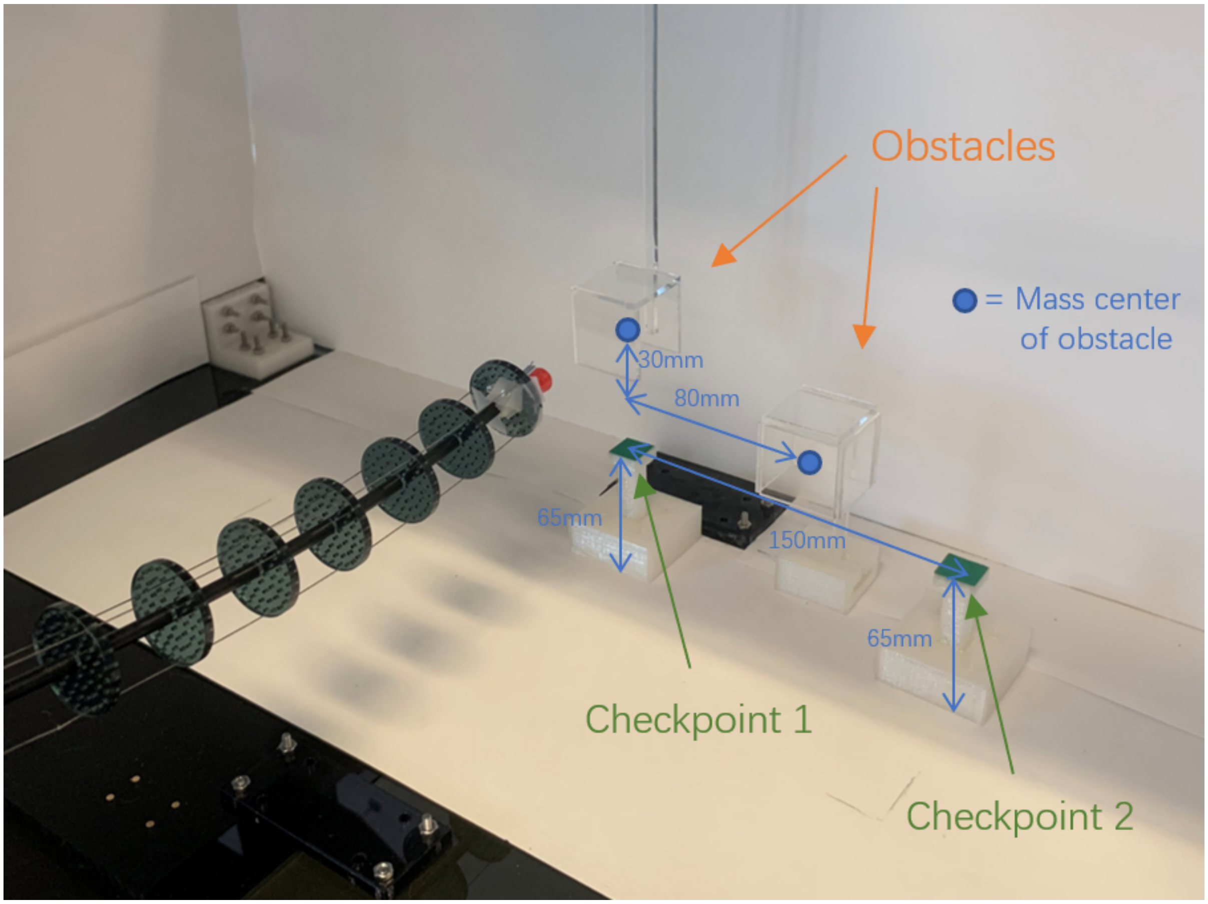

Figure 4.4. The layout of avoiding obstacle task.

Force feedback in this task is simply $F = k\Delta x$, where k=500 N/m and Δx is as shown in figure 4.5. We set a virtual wall around the obstacle when we calculate the force feedback, and Δx is how long the robot tip penetrate the virtual wall. If the robot tip is in the range of virtual wall, there will be resisting force feedback. The existence of the virtual wall is to make sure that the user can feel force feedback when the robot tip is near the obstacle before it actually hits the obstacle, so the force feedback can be a warning to avoid obstacles.

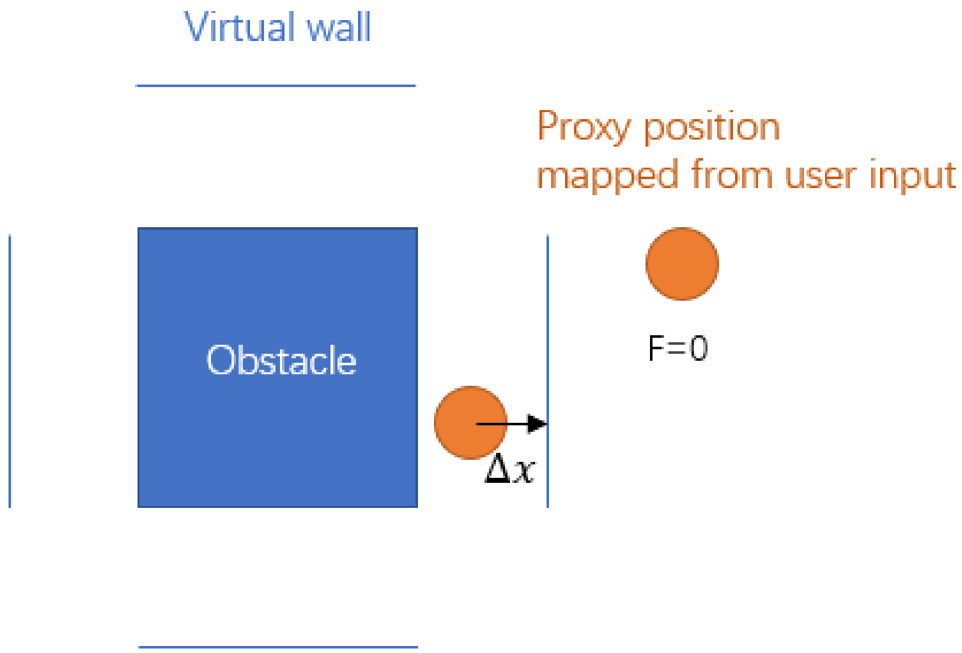

Figure 4.5. The force feedback in avoiding obstacle task.

4.1.5 Other preparations for the study

The order of different device setups and tasks should not be the same across participants. According to the principle of latin-square design, we should avoid the potential learning effect brought by any specific order of operations or conditions [9]. The latin-square design basically requires all possible orders of conditions to take place an equal number of times within the whole experiment for all the users. The order of tasks and device setups in our study is therefore designed based on this:

Table 4.1. Order of tasks and device setups. For device setups, A = trackball with force feedback, B = Trackball without force feedback, C = Joystick.

User ID	Task 1 (Device order)	Task 2 (Device order)
1	Following paths (A,B,C)	Avoiding obstacles (C,A,B)
2	Avoiding obstacles (A,B,C)	Following paths (C,A,B)
3	Following paths (C,A,B)	Avoiding obstacles (B,C,A)
4	Avoiding obstacles (C,A,B)	Following paths (B,C,A)
5	Following paths (B,C,A)	Avoiding obstacles (A,B,C)
6	Avoiding obstacles (B,C,A)	Following paths (A,B,C)
7	Following paths (A,B,C)	Avoiding obstacles (C,A,B)
8	Avoiding obstacles (A,B,C)	Following paths (C,A,B)
9	Following paths (C,A,B)	Avoiding obstacles (B,C,A)
10	Avoiding obstacles (C,A,B)	Following paths (B,C,A)
11	Following paths (B,C,A)	Avoiding obstacles (A,B,C)
12	Avoiding obstacles (B,C,A)	Following paths (A,B,C)

After users complete two tasks, they will be asked to fill out a post-study survey. The survey contains 3 kinds of questions. (1) Rating questions: rate each device setup in terms of precision, speed control, mental effort required and smoothness during navigation. (2) Ranking questions: rank 3 device in different tasks. (3) Yes-or-no question: whether you have experience with video games, 3D controllers and whether the force feedback is clear and helpful.

Due to the spreading of Covid-19 at the moment when the study is conducted, several extra preparations are considered in this study. The user will be asked to do a prescreening over phone the day before the study takes place to confirm they have no symptoms or positive tests results. PPEs including masks are prepared, and all the devices will be sanitized before the study starts.

4.2 Experiment setup

Users will be sitting aside the tendon-driven robot and use different interfaces to teleoperate it. Figure 4.6 shows how users operating the tendon-driven robot to complete tasks in the workspace.

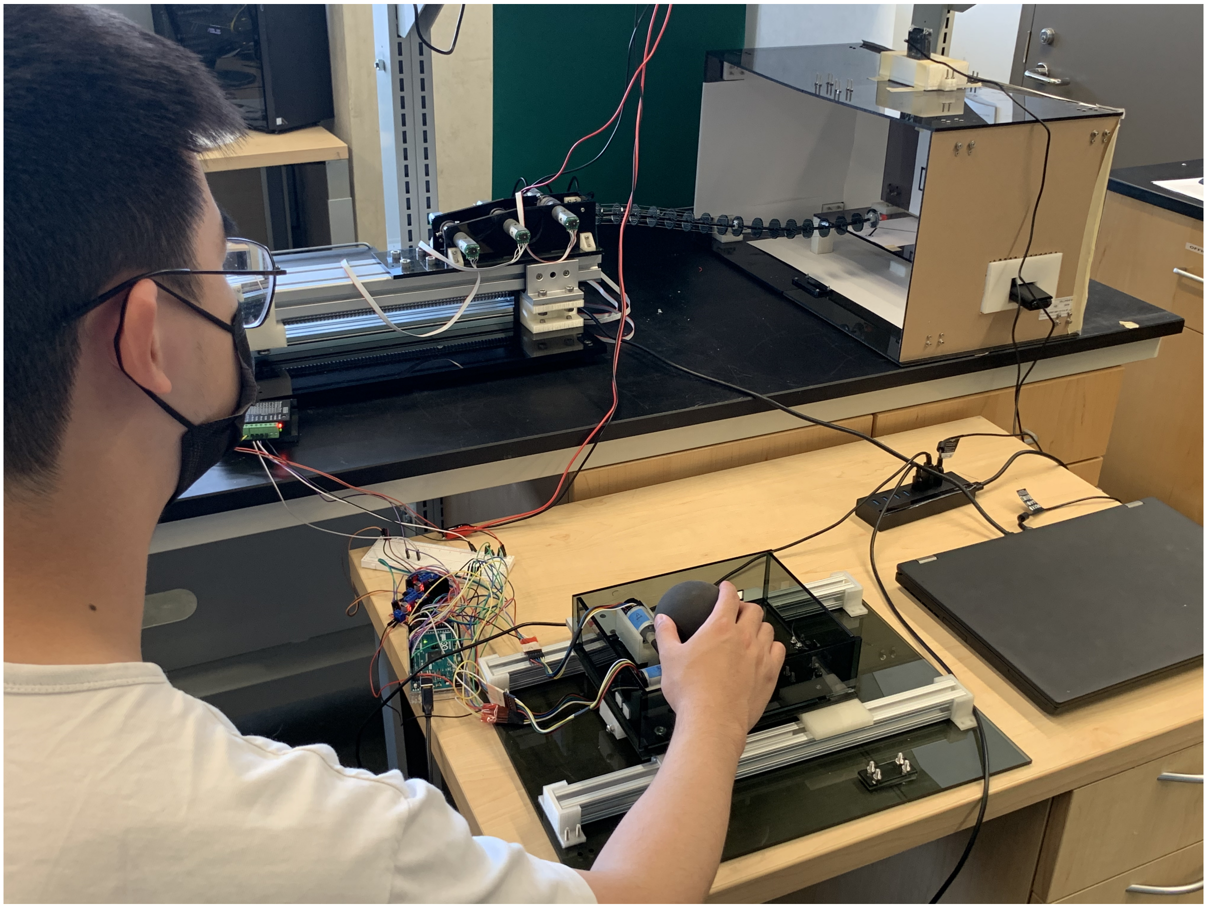

Figure 4.6. A user teleoperating the robot with our proposed interface.

All the tasks are completed in a $60 \cdot 30 \cdot 30cm$ workspace, built using acrylic boards. There are two cameras, one on the top roof and the other on the right wall, to track the 3D position of the robot tip in real time. The inner surface is covered with white cardboard, so the red label at robot tip can be clearly recognized by the tracking algorithm.

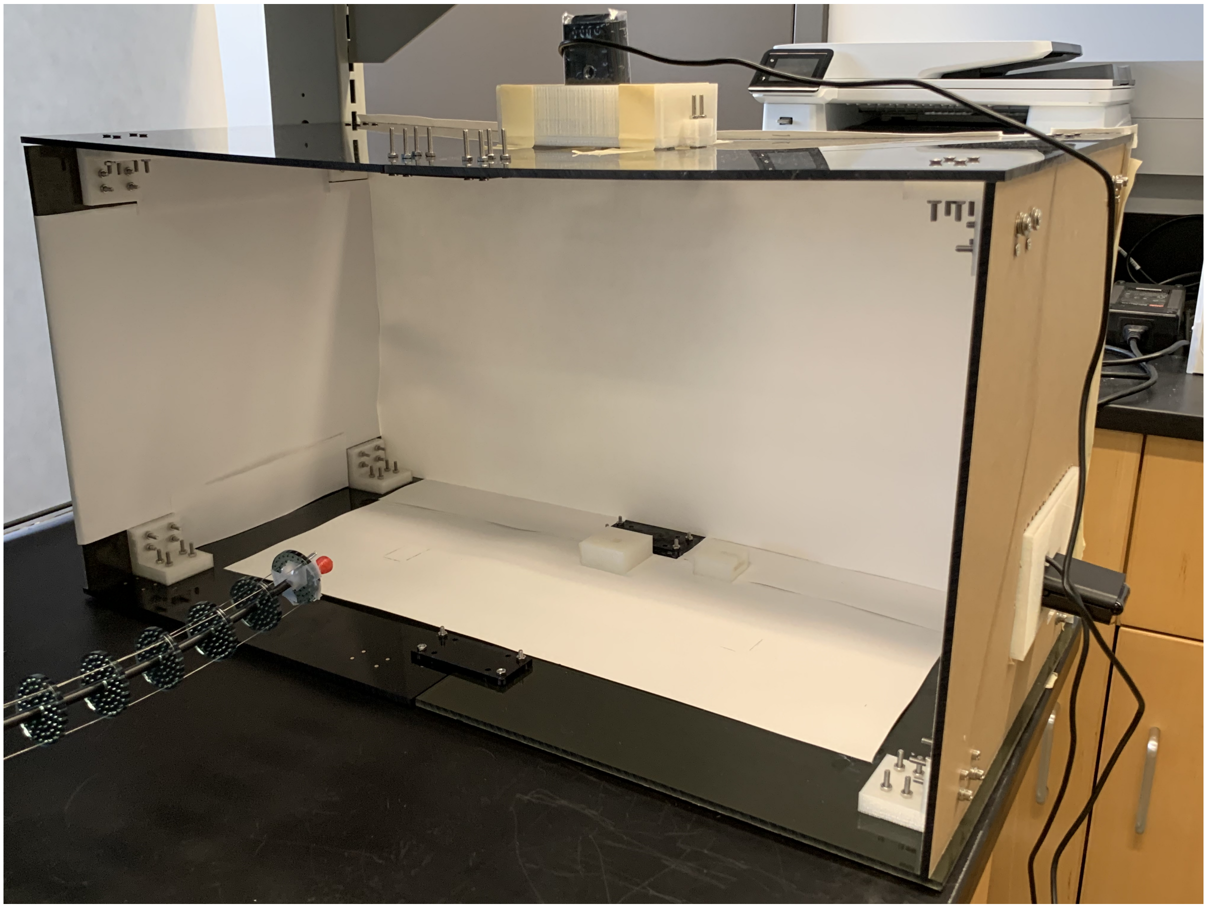

Figure 4.7. The workspace. Two cameras are mounted on the right wall and top roof.

The setup for the two tasks is different, so we will change the setup between the tasks. All the parts including paths for following, obstacles and checkpoints are self-made, and they are designed to be easily assembled and disassembled.

The joystick we use for comparison is an off-the-shelf gamepad joystick, Logitech F310. For fair comparison, the joystick is designed to be P2P mapping on the x-axis and y-axis, similar to the trackball interface. Since the deployment on the z-axis is relatively long, and position mapping may not apply for such joystick mechanisms, we make the mapping on the z-axis a velocity mapping. This mapping, although not consistent on the joystick setup, remains part of the nature of joysticks and keeps the comparison fair to some degree. Also, it still turns out to be intuitive for users based on the pilot tests conducted before the official user study.

The overall mapping is shown as follows, where arrows in different colors shows the direction of x, y and z axis in the robot frame, joystick and trackball interface.

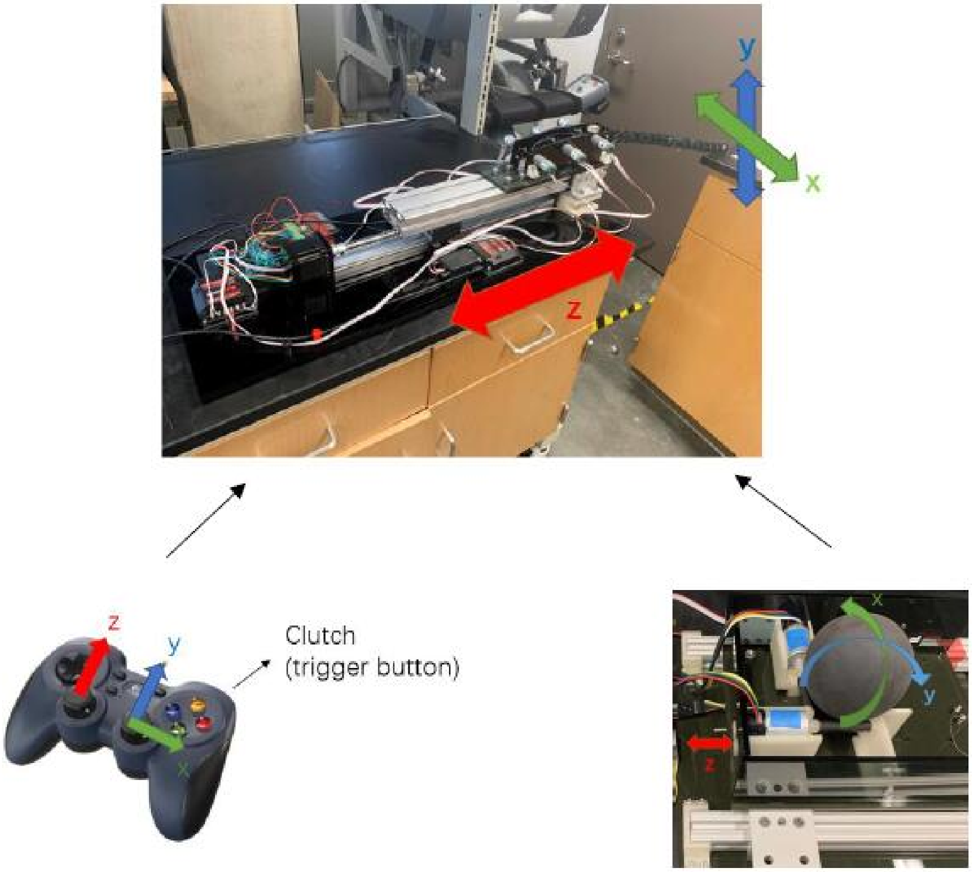

Figure 4.8. The mapping of the system. On the top is the frame of motion of the tendon-driven robot. Bottom left and bottom right show the mapping of joystick and trackball interface.

A special design of the joystick is a clutching button. This function is necessary for two reasons. First, the range of the joystick is small. Second, the joystick has a return spring, which makes the robot move back to the origin once the user removes their finger from the joystick, if a P2P mapping is used. We would like to avoid this situation and extend the mapping by adding a clutching function. When the user pushes the joystick to any position and presses the clutching button, the robot will be re-centered at the current position. Loosening the clutching button and letting the joystick go back to the origin again will control the robot to start from the re-centered position. The clutching button is set to be the right trigger button on the gamepad, considered to be a comfortable gesture for most right-handed users.

4.3 Evaluation metrics

The evaluation metrics include two parts: quantitative analysis and qualitative analysis. For the path-following task, the quantitative data is the runtime and the error between users' paths and the given path. The corresponding time of each point on the actual trajectory is recorded along with the actual trajectory vector. The error is defined as the shortest distance between each point on the actual trajectory and the given path (the normal distance between a point and a line). From the study, we record a 6*N vector for each trial of following paths task (N is the number of points on the actual trajectory):

$$data = \begin{bmatrix} x_1 & x_2 & ... & x_N \\ y_1 & y_2 & ... & y_N \\ z_1 & z_2 & ... & z_N \\ h_1 & h_2 & ... & h_N \\ m_1 & m_2 & ... & m_N \\ s_1 & s_2 & ... & m_N \end{bmatrix} \tag{4.1}$$

where (x,y,z) corresponds to the actual trajectory of the user and (h,m,s) corresponds to the time (hour, minute, second) of each point. We can use (x,y,z) to calculate the error and (h,m,s) to calculate the runtime.

For the obstacle-avoidance task, the quantitative data is the runtime and the number of times that the obstacles were hit. The runtime is computed in a similar way, and the number of times hitting obstacles is recorded by the instructor of the study, then further verified by the actual trajectory afterwards.

The qualitative data is a post-study survey, as described in section 4.1. The data from the survey is also subjected to all the statistical analysis, as shown in the following section.

4.4 Study results

We have 12 participants for the study, each completing the two tasks with 3 different device setups in 3 repeated trials. Among the 12 participants, there are 3 females and 9 males, with an average age of 23.6 ± 2.9. All of them are right-handed. For the following paths task, we calculate the error mean $\bar{E}$ for each device setup.

$$\bar{E} = \frac{1}{12}\sum_{i=1}^{12} E_i \tag{4.2}$$

E_i is the error mean of the i-th user, averaged from error mean in three trials:

$$E_i = \frac{1}{3}\sum_{j=1}^{3} E_{ij} \tag{4.3}$$

E_{ij} is computed from the actual trajectory data:

$$E_{ij} = \frac{1}{N}\sum_{n=1}^{N} D(\mathbf{X_n}, \mathbf{P}) \tag{4.4}$$

where $\mathbf{X_n}$ is a point on the actual trajectory, $\mathbf{P}$ is the given path to follow, and $D(\mathbf{X_n}, \mathbf{P})$ is the shortest distance from a point $\mathbf{X_n}$ to a vector line $\mathbf{P}$. The boxplot of error mean is shown below.

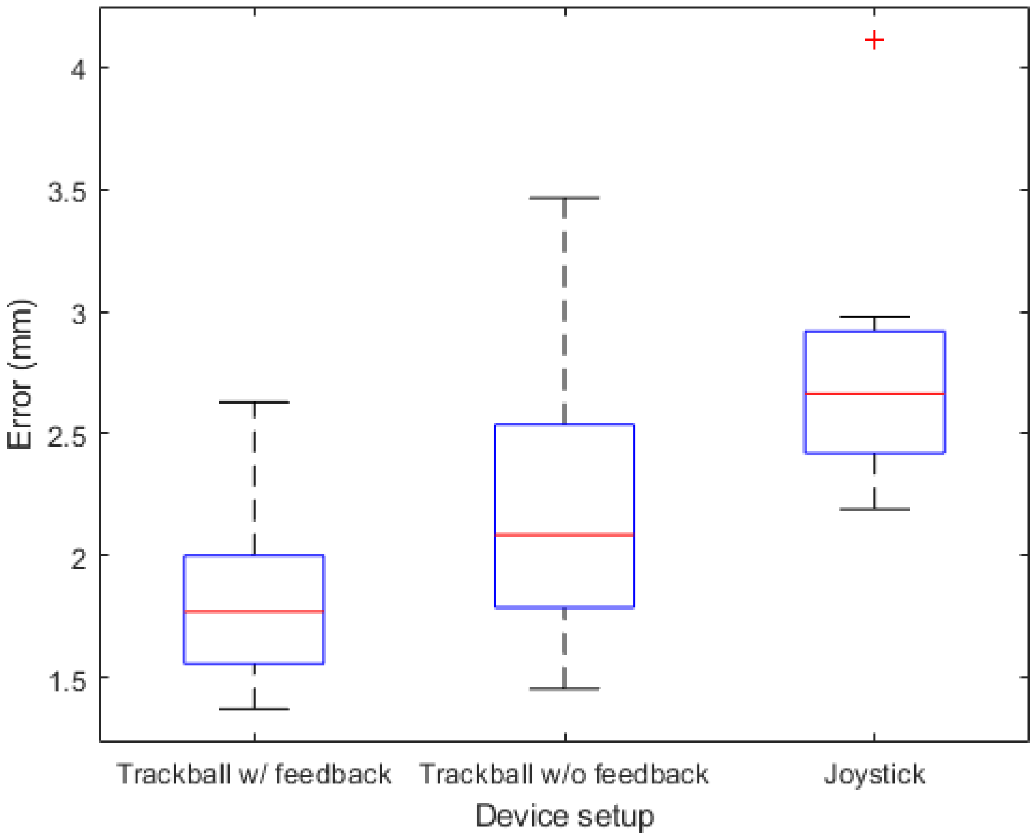

Figure 4.9. The box plot of errors in following paths task.

To find the level of confidence for significant difference of 3 device setups, we conduct a Kruskal-Wallis test and find p-value is 0.0011, which indicates a 99% confidence level of significant difference among 3 device setups. To find specifically which device setup is significantly different than others, we conduct a Mann-Whitney test with Bonferroni correction between each pairs of two device setups. The p-value between trackball with feedback and trackball without feedback is 0.33. The p-value between trackball with feedback and joystick is 0.0012. The p-value between trackball without feedback and joystick is 0.066. From the above analysis, we can conclude that under 95% confidence level, in terms of precision, the performance of the trackball with force feedback is significantly different from the joystick.

These are some representative examples of actual trajectories of the robot tip in the first part of the path. In each plot, the curved lines in different colors are the actual trajectories of 3 trials, the solid red lines are the center line of the path, and the dashed red lines are the width of tolerance, which is the boundaries for force feedback. From the plots we can intuitively verify that trackball with force feedback has the highest precision among all device setups.

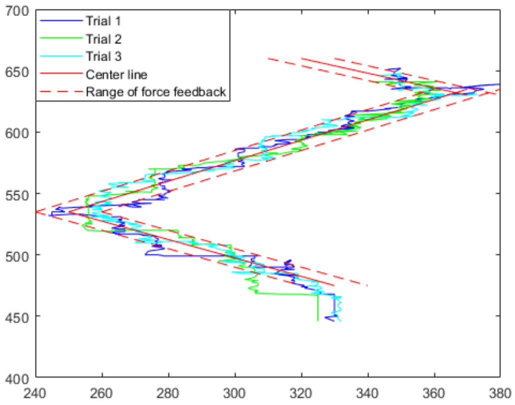

Figure 4.10. The actual paths using trackball with force feedback.

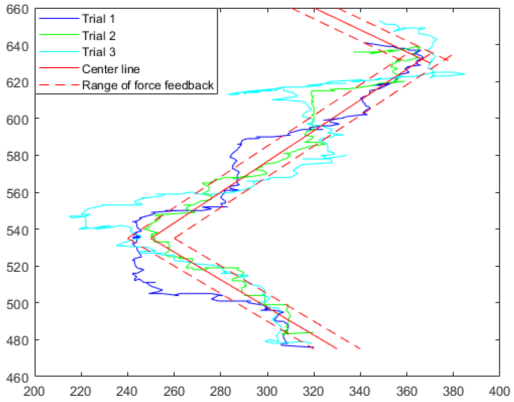

Figure 4.11. The actual paths using trackball without force feedback.

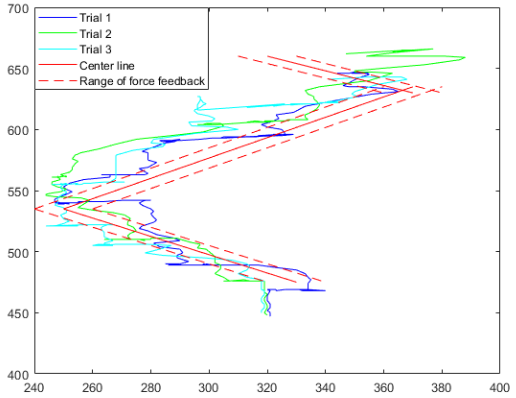

Figure 4.12. The actual paths using joystick.

The runtime of following paths task is also averaged in a similar way. The boxplot of runtime is shown below.

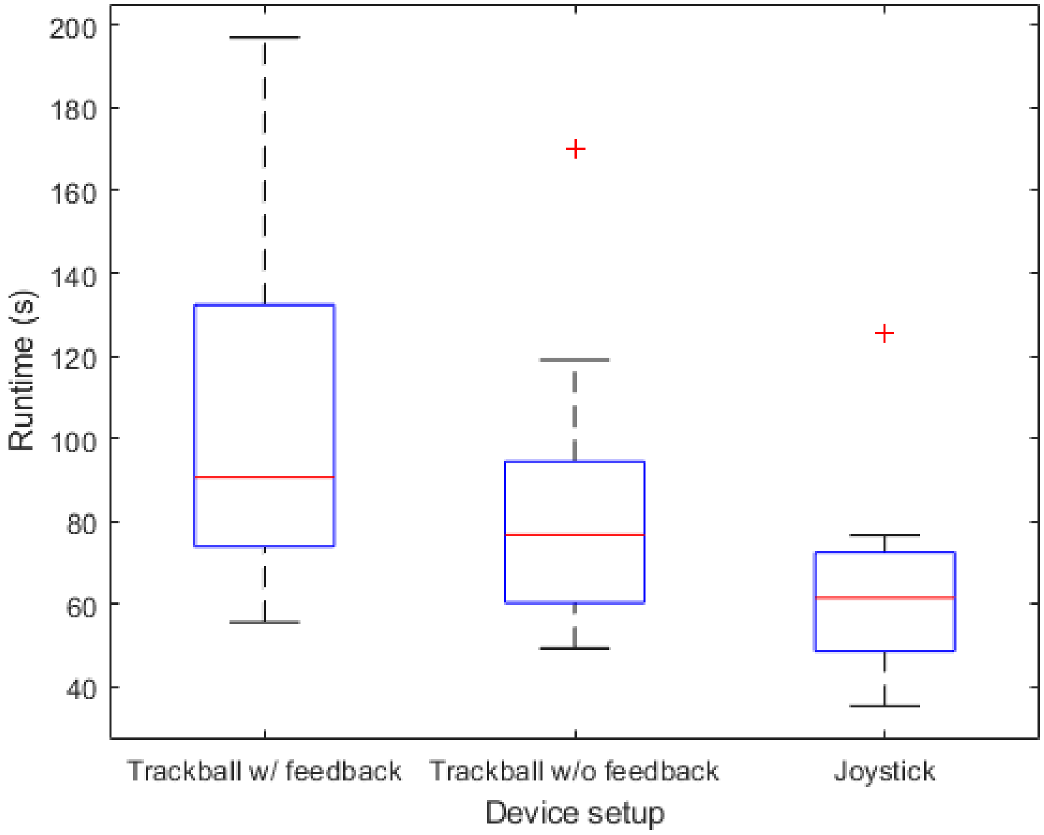

Figure 4.13. The box plot of runtime in following paths task.

We also conduct statistical analysis for the runtime. The p-value from Kruskal-Wallis test is 0.013, which indicates a 95% confidence level of significant difference among 3 device setups. The p-value between trackball with feedback and trackball without feedback is 0.51. The p-value between trackball with feedback and joystick is 0.012. The p-value between trackball without feedback and joystick is 0.33. From the above analysis, we can conclude that under 95% confidence level, in terms of runtime, the trackball with force feedback is significantly different from the joystick.

Quantitative results for following paths task are as follows.

Table 4.2. Error (mm) and runtime (s) of 3 device setups in following paths task.

	Trackball with force feedback	Trackball without force feedback	Joystick
Error Avg.	1.85	2.20	2.74
Error Std.	0.41	0.58	0.50
Time Avg.	106.09	83.83	63.72
Time Std.	45.04	34.11	23.63

For the obstacle-avoidance task, we record the number of times that the robot tip hits obstacles in each trial and compute the average. From another perspective, the data shows that 75% of users did not hit any obstacle during the three trials for the trackball with force feedback. For the trackball without force feedback, the result is 58%, and for the joystick it is 50%. The results show that the trackball with force feedback performs the best in avoiding obstacles.

Similarly, the results of runtime are also shown in box plot.

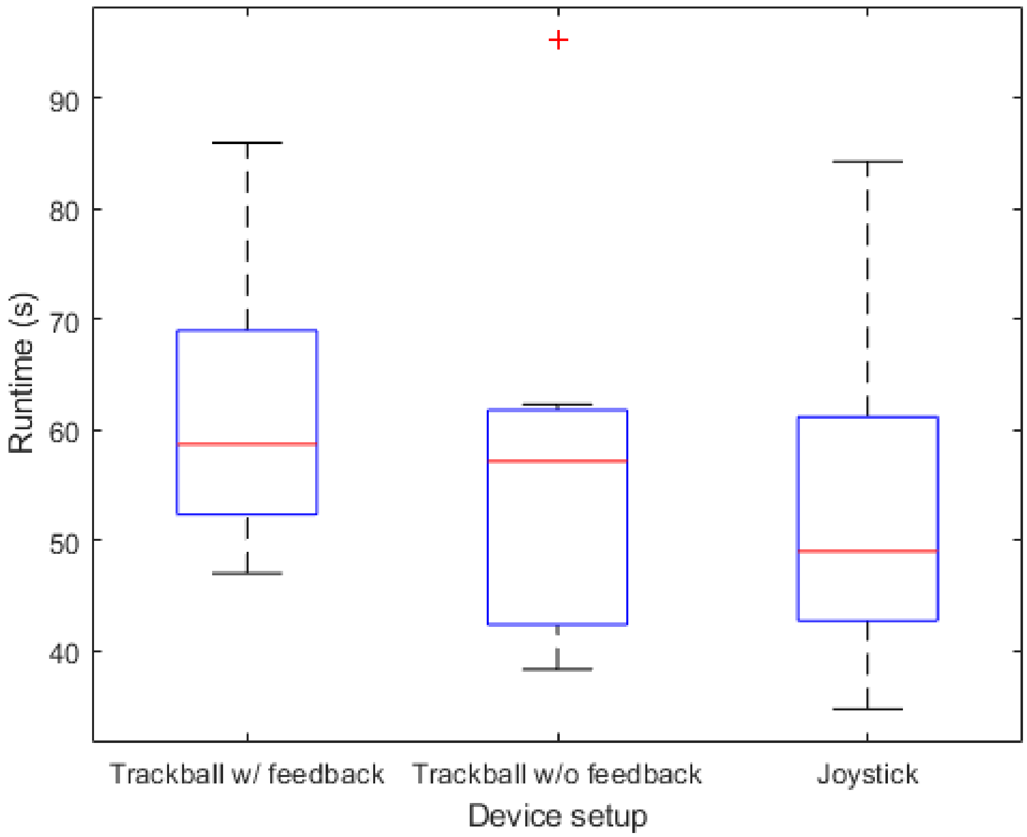

Figure 4.14. The box plot of runtime in avoiding obstacles task.

Quantitative results for Avoiding obstacles task are as follows.

Table 4.3. Average times hitting obstacles in each trial and runtime of avoiding obstacles task.

	Trackball with force feedback	Trackball without force feedback	Joystick
Hit Avg.	0.08	0.22	0.28
Time Avg.	61.54	55.51	52.42
Time Std.	12.79	15.54	14.44

The Kruskal-Wallis test gives a p-value of 0.249, which indicates that no significant difference of runtime is observed among three device setups.

Results of runtime in both tasks indicate that our 3D haptic trackball interface is more time-consuming than conventional joystick interfaces. This is expected since the force feedback may prevent the robot move in a wrong direction, thus add some time for correction. Moreover, it may make the user being more cautious after they feel the resisting force, which also slows down the teleoperation naturally. More importantly, the runtime is likely a result of the slow loop rate, due to the serial port communications.

As for the post-study survey, we first ask whether the user has experience with video games and input devices with at least 3 DOFs. The answer is 9 out of 12 users have such experience. This shows that most of our users know the basics of hand-eye coordination. We also ask whether the force feedback is clearly felt and helpful when controlling the robot. 8 users think the force feedback is assisting them with controlling the robot. We then ask users to rate and rank the three device setups in different aspects. The results are as follows.

Table 4.4. Average ratings of 3 device setups in different aspects, with 5 being the best (or most mental effort) and 1 being the worst (or least mental effort).

Rating aspect	Trackball with force feedback	Trackball without force feedback	Joystick
Precision	3.83	3.92	2.83
Speed control	3.83	3.83	3.5
Mental effort required	3.42	3.25	2.83
Smoothness during navigation	3.75	4.17	2.75
General comfort	3.50	3.58	3.17

Table 4.5. Average ranking of 3 device setups in different aspects, with 1 being the first rank and 3 being the lowest rank.

Ranking aspect	Trackball with force feedback	Trackball without force feedback	Joystick
In following paths task	1.58	1.42	3
In Avoiding obstacles task	1.83	2.08	2.08
For teleoperating continuum robots	1.58	1.58	2.83

We also perform analysis for these subjective answers. The results shows that in general, users think trackball with and without force feedback are performing similarly well, and they are both better than the joystick. The only exception is for the question asking about the mental effort required. Users clearly rate the trackball with force feedback the most mental-effort-consuming interface, and the joystick is the least. This is reasonable in that users have to feel the force feedback and react to it while teleoperating the robot, so more mental effort is needed in this scenario. In contrast, since most users have experience in video games, they will be more familiar with the joystick on gamepad, thus spending less mental efforts on that.

Chapter 5

Conclusions and Future Work

We developed a novel 3D haptic trackball interface for controlling tendon-driven continuum robots. The interface was able to teleoperate the tip position of the robot in 3D space and provide haptic force feedback to users to assist them with controlling the robot. We also performed a user study including a training section and two specific tasks to evaluate the performance of our interface. In the study, an off-the-shelf joystick interface was used to perform the same task as comparison. The cost of the whole system, including hardware on the tendon-driven robot, the trackball and joystick, motors, controllers and cameras, was around \$700 for the prototype. The cost should be reduced significantly if the prototype is commercialized, since most of the hardware could be purchased with a lower price in a large amount.

The advantage of our interface is the haptic force feedback, which improves the accuracy of teleoperation significantly and guarantees the safety in terms of avoiding hitting obstacles. The high accuracy is benefited from both the nature of trackball P2P mapping and the presence of haptic feedback. These benefits are clearly shown in the results of user study. The haptic trackball interface has the best performance in both precisely following a given path and avoiding obstacles during navigation. The fact that the trackball interface, even without haptic feedback, still performs better than the joystick in these aspects, shows the advantage of trackball structure over conventional joystick. It should be noted that in some cases, the trackball interface has a similar performance with and without haptic feedback. This is because the task is designed to be not difficult, and some users with rich experience in teleoperation are able to perform the task well enough without the help of haptic feedback (i.e., they can almost perfectly follow the path and avoid obstacles). In other cases, some users are less experienced in teleoperation, so the presence of haptic feedback can improve their performance in the tasks. Therefore, it can be expected that the advantage of our interface will be more obvious in more difficult tasks.

It is important to realize some limitations of our current design. First, from the results of user study, we can see that the haptic trackball interface is the most time-consuming one in both tasks. This drawback, although has a reduced influence in cases where speed is not an important factor, can still be improved for better user experience. The long runtime is partially resulted from the presence of haptic feedback, since users need extra time to notice the force feedback and react to it, but more importantly, limited by the speed of serial communications. Considering convenience in motor control and code implementation, we use Arduino to control the motors in the interface and MATLAB for main control loop, so serial communications between them are inevitable, which bring some delays in the loop. Disregarding the choice of motors and controllers, we should be able to integrate all the coding and control in one platform, thus there is room for higher loop rate if we are to commercialize the system.

Second, our system is only capable of controlling tip position of the robot. We have no body shape optimization of the tendon-driven continuum robot. Our current inverse kinematics is based on an initial guess of arc parameters at the previous moment, so it is always solving for a set of arc parameters which is close to the previous moment. Therefore, we have no control on the body shape and are not able to prevent the body from hitting obstacles. This is an important reason why we can not design a difficult task in the user study. If we want to control the body shape as well, the interface should also be adjusted to have more DOFs. An algorithm of optimizing the body shape in inverse kinematics can also work according to the content of specific tasks, if we still remain 3 DOFs on the interface.

Finally, optimization of design parameters is always a direction for future work. For example, we can choose better materials for the trackball and motor shafts so that friction force is larger and the teleoperation can go more smoothly. We can choose motors with larger torque to provide larger forces. We can also use lighter materials for manufacture so the system can become easily portable. There are still a large room for our prototype to be commercialized, but we can already see most of its benefits now.

This new interface also has potential to be used as a leader device for other robots. For example, continuum robots in general can fit well with this device, as long as we aim to control the tip position of the robot. As mentioned before, the inverse kinematics is robot-independent, so we can directly use the implemented algorithm to solve for arc parameters. However, the robot-specific control of the motors varies for different types of continuum robots, and the haptic feedback can always be adapted according to the tasks of interests.

Bibliography

[1] V. P. Vega A. J. Silva, O. A. D. Ramirez and J. P. O. Oliver. Phantom omni haptic device: Kinematic and manipulability. *2009 Electronics, Robotics and Automotive Mechanics Conference (CERMA)*, pages 193–198, 2009.

[2] Burgner-Kahrs J. Amanov E, Nguyen T-D. Tendon-driven continuum robots with extensible sections - a model-based evaluation of path-following motions. *The International Journal of Robotics Research*, 40(1):7–23, 2021.

[3] Mali Shen-Mohamed EMK Abdelaziz Burak Temelkuran Anzhu Gao, Ning Liu and Guang-Zhong Yang. Laser-profiled continuum robot with integrated tension sensing for simultaneous shape and tip force estimation. *Soft Robotics*, 7(4):421–443, Aug 2020.

[4] H. Tam B. Ouyang, Y. Liu and D. Sun. Design of an interactive control system for a multisection continuum robot. *IEEE/ASME Transactions on Mechatronics*, 23(5):2379–2389, Oct 2018.

[5] Hirose S. Breedveld, P. Design of steerable endoscopes to improve the visual perception of depth during laparoscopic surgery. *ASME. J. Mech. Des.*, 126(1):2–5, Jan 2004.

[6] Jessica Burgner-Kahrs Carolin Fellmann, Daryoush Kashi. Evaluation of input devices for teleoperation of concentric tube continuum robots for surgical tasks. *Medical Imaging 2015: Image-Guided Procedures, Robotic Interventions, and Modeling*, 2015.

[7] Mohamad S. Hamady Celia V. Riga, Colin D. Bicknell and Nicholas J.W. Cheshire. Evaluation of robotic endovascular catheters for arch vessel cannulation. *Journal of Vascular Surgery*, 54(3):799–809, 2011.

[8] Peng Qi Tianliang Li Shuang Song Zoran Najdovski-Toshio Fukuda Chaoyang Shi, Xiongbiao Luo and Hongliang Ren. Shape sensing techniques for continuum robots in minimally invasive surgery: A survey. *IEEE Transactions on Biomedical Engineering*, 64(8):1665–1678, Aug 2017.

[9] N. W Chilton. The latin square design in clinical experimentation. *Journal of dental research*, 34(3):421–428, 1955.

[10] M. Dehghani and S. A. A. Moosavian. Modeling and control of a planar continuum robot. *2011 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 966–971, 2011.

[11] N. Diolaiti and C. Melchiorri. Tele-operation of a mobile robot through haptic feedback. *IEEE International Workshop HAVE Haptic Virtual Environments and Their*, pages 67–72, 2002.

[12] I. D. Walker E. Tatlicioglu and D. M. Dawson. New dynamic models for planar extensible continuum robot manipulators. *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1485–1490, 2007.

[13] Iwata H. History of haptic interface. *Human Haptic Perception: Basics and Applications*, 2008.

[14] Zain Mehdi Sang-Goo Jeong Muhammad Usman-Elliot W. Hawkes Allison M. Okarnura Haitham El-Hussieny, Usman Mehmood and Jee-Hwan Ryu. Development and evaluation of an intuitive flexible interface for teleoperating soft growing robots. *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4995–5002, 2018.

[15] Yanfei Cao Fei Qi-Dongming Bai Yaoyao Wang Hao Guo, Feng Ju and Bai Chen. Continuum robot shape estimation using permanent magnets and magnetic sensors. *Sensors and Actuators A: Physical*, 285:519–530, 2019.

[16] Li Q Xie H-Shen R He B, Wang Z. An analytic method for the kinematics and dynamics of a multiple-backbone continuum robot. *International Journal of Advanced Robotic Systems*, Jan 2013.

[17] Tamar Flash Frank W. Grasso Roger T. Hanlon Binyamin Hochner William M. Kier Christopher C. Pagano Christopher D. Rahn Ian D.Walker, Darren M. Dawson and Qiming M. Zhang. Continuum robot arms inspired by cephalopods. *Unmanned Ground Vehicle Technology VII*, May 2005.

[18] D. C. Rucker J. Burgner-Kahrs and H. Choset. Continuum robots for medical applications: A survey. *IEEE Transactions on Robotics*, 31(6):1261–1280, Dec 2015.

[19] T. Okumura I. Kose H. Takagi K. Kato, Kato and N. T. Hata. Tendon-driven continuum robot for neuroendoscopy: validation of extended kinematic mapping for hysteresis operation. *International Journal of Computer Assisted Radiology and Surgery*, 11(4):589–602, 2016.

[20] Soheil Kianzad and Karon E. MacLean. Harold's purple crayon rendered in haptics: Large-stroke, handheld ballpoint force feedback. *2018 IEEE Haptics Symposium (HAPTICS)*, 2018.

[21] Martin CW Leong Marco Chow Hing-Choi Fu Kaspar Althoefer Kam Yim Sze Chung-Kwong Yeung Kit-Hang Lee, Denny KC Fu and Ka-Wai Kwok. Nonparametric online learning control for soft continuum robot: An enabling technique for effective endoscopic navigation. *Soft Robotics*, 4(4):324–337, Dec 2017.

[22] W. McMahan V. Iyengar M. Csencsits, B. A. Jones and I. D. Walker. User interfaces for continuum robot arms. *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3123–3130, 2005.

[23] Thomas Harold Massie. Design of a three degree of freedom force-reflecting haptic interface. *Diss. Massachusetts Institute of Technology*, 1993.

[24] Robert J. Webster Murphy, Todd E. and Allison M. Okamura. Design and performance of a two-dimensional tactile slip display. *Proc. Eurohaptics*, 2004.

[25] S. Neppalli and B. A. Jones. Design, construction, and analysis of a continuum robot. *007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1503–1507, 2007.

[26] Lau D Ng K.W., Mahony R. A dual joystick-trackball interface for accurate and time-efficient teleoperation of cable-driven parallel robots within large workspaces. *Cable-Driven Parallel Robots. CableCon 2019. Mechanisms and Machine Science*, 74, 2019.

[27] B. Itkowitz P. E. Dupont, J. Lock and E. Butler. Design and control of concentric-tube robots. *IEEE Transactions on Robotics*, 26(2):209–225, April 2010.

[28] G. Robinson and J. B. C. Davies. Continuum robots—a state of the art. *Proceedings 1999 IEEE International Conference on Robotics and Automation*, 4:2849–2854, 1999.

[29] Vikas Rao Sayan Seth Sourajit Mukherjee Aparajita Sengupta-Subhasis Bhaumik Saptak Bhattacherjee, Sudipta Chattopadhayay. Kinematics and teleoperation of tendon driven continuum robot. *Procedia Computer Science*, 133:879–886, 2018.

[30] P. Sears and P. Dupont. A steerable needle technology using curved concentric tubes. *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2850–2856, 2006.

[31] A. Hildebrandt T. Mahl and O. Sawodny. variable curvature continuum kinematics for kinematic control of the bionic handling assistant. *JIEEE Transactions on Robotics*, 30(4):935–949, Aug 2014.

[32] Weaver K.D. Webster III R.J Travaglini T.A., Swaney P.J. Initial experiments with the leap motion as a user interface in robotic endonasal surgery. *Mechanisms and Machine Science*, 37, 2016.

[33] Robert J. Webster III and Bryan A. Jones. Design and kinematic modeling of constant curvature continuum robots: A review. *The International Journal of Robotics Research*, 29(13):1661–1683, 2010.

[34] S. S. Cheng Y. Kim and J. P. Desai. Active stiffness tuning of a spring-based continuum robot for mri-guided neurosurgery. *IEEE Transactions on Robotics*, 34(1):18–28, Feb 2018.

[35] Takeo Kanade Yasuyoshi Yokokohji, Ralph L. Hollis. Wysiwyf display: A visual/haptic interface to virtual environment. *Teleoperators and Virtual Environments*, 8(4):412–434, 1999.

[36] Jaydev P. Desai Yeongjin Kim, Shing Shin Cheng. Towards the development of a spring-based continuum robot for neurosurgery.

[37] Yi B-J Yoon H-S. Design of a master device for controlling multi-moduled continuum robots. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 231(10):1921–1931, 2017.

[38] Q. Zhao and F. Gao. Design and analysis of a kind of biomimetic continuum robot. *010 IEEE International Conference on Robotics and Biomimetics*, pages 1316–1320, 2010.